

Advances in Security and Performance of Blockchain Systems

Lead Guest Editor: Yuling Chen

Guest Editors: Xinyi Huang, Xiaodong Lin, and Xiu-Bo Chen





Advances in Security and Performance of Blockchain Systems

Security and Communication Networks

Advances in Security and Performance of Blockchain Systems

Lead Guest Editor: Yuling Chen

Guest Editors: Xinyi Huang, Xiaodong Lin, and
Xiu-Bo Chen







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
Vicente 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

A New Lattice-Based Blind Ring Signature for Completely Anonymous Blockchain Transaction Systems

Yi-Yang Xie, Xiu-Bo Chen , and Yi-Xian Yang



Research Article (12 pages), Article ID 4052029, Volume 2022 (2022)

A Privacy-Aware Electricity Consumption Data Collection Model Based on Group Blind Signature

Fengyin Li , Xiao Li , Peiyu Liu , Xueqing Sun , Siqi Yu , and Junrong Ge 

Research Article (14 pages), Article ID 4352291, Volume 2022 (2022)

IIDQN: An Incentive Improved DQN Algorithm in EBSN Recommender System

Jianan Guo , Yilei Wang , Hui An, Ming Liu, Yiting Zhang, and Chunmei Li

Research Article (12 pages), Article ID 7502248, Volume 2022 (2022)

Cloud Storage Data Access Control Scheme Based on Blockchain and Attribute-Based Encryption

Xiaodong Yang , Aijia Chen , Zhisong Wang, and Shudong Li 

Research Article (12 pages), Article ID 2204832, Volume 2022 (2022)

Multiparty Data Publishing via Blockchain and Differential Privacy

Zhen Gu , Kejia Zhang , and Guoyin Zhang 

Research Article (13 pages), Article ID 5612794, Volume 2022 (2022)

An Edge Cloud Data Integrity Protection Scheme Based on Blockchain

Weihua Duan , Yu Jiang , Xiaolong Xu , Ziming Zhang , and Guanpei Liu 



Research Article (15 pages), Article ID 5016809, Volume 2022 (2022)

BSD-Guard: A Collaborative Blockchain-Based Approach for Detection and Mitigation of SDN-Targeted DDoS Attacks

Shanqing Jiang , Lin Yang , Xianming Gao , Yuyang Zhou , Tao Feng , Yanbo Song , Kexian Liu , and Guang Cheng 

Research Article (16 pages), Article ID 1608689, Volume 2022 (2022)

A Noninteractive Multireplica Provable Data Possession Scheme Based on Smart Contract

Zhengwen Li , Yang Xin, De Zhao , and Yixian Yang



Research Article (14 pages), Article ID 6268449, Volume 2022 (2022)

Blockchain Data Sharing Query Scheme based on Threshold Secret Sharing

Lu Chen , Xin Zhang , and Zhixin Sun 

Research Article (12 pages), Article ID 8996815, Volume 2022 (2022)

A Blockchain-Based User Authentication Scheme with Access Control for Telehealth Systems

Shuyun Shi, Min Luo , Yihong Wen, Lianhai Wang, and Debiao He 




Research Article (18 pages), Article ID 6735003, Volume 2022 (2022)

CTRF: Ethereum-Based Ponzi Contract Identification


Xuezhi He , Tan Yang , and Liping Chen 

Research Article (10 pages), Article ID 1554752, Volume 2022 (2022)






PPSEB: A Postquantum Public-Key Searchable Encryption Scheme on Blockchain for E-Healthcare Scenarios

Gang Xu , Shiyuan Xu , Yibo Cao, Fan Yun, Yu Cui, Yiyang Yu, and Ke Xiao 
Research Article (13 pages), Article ID 3368819, Volume 2022 (2022)

Research on the Millionaires' Problem under the Malicious Model Based on the Elliptic Curve Cryptography

Xin Liu, Yang Xu, Gang Xu , and Baoshan Li
Research Article (11 pages), Article ID 6923610, Volume 2022 (2022)




A Privacy Protection Scheme for Facial Recognition and Resolution Based on Edge Computing

Junhua wu , Wenzhen Feng , Guopeng Liang , Tiantian Wang , Guangshun Li , and Yuanwang Zheng
Research Article (12 pages), Article ID 4095427, Volume 2022 (2022)

ATMChain: Blockchain-Based Security Framework for Cyber-Physics System in Air Traffic Management

Xin Lu  and Zhijun Wu 
Research Article (11 pages), Article ID 8542876, Volume 2022 (2022)




Privacy-Preserving Minority Oversampling Protocols with Fully Homomorphic Encryption

Maohua Sun , Ruidi Yang , and Mengying Liu 
Research Article (9 pages), Article ID 3068199, Volume 2022 (2022)

Privacy-Preserving Collaborative Computation for Human Activity Recognition

Lin Wang, Chuan Zhao , Kun Zhao , Bo Zhang , Shan Jing, Zhenxiang Chen, and Kuiheng Sun
Research Article (8 pages), Article ID 9428610, Volume 2022 (2022)


Multiple-Layer Security Threats on the Ethereum Blockchain and Their Countermeasures

Li Duan , Yangyang Sun , Kejia Zhang , and Yong Ding 
Research Article (11 pages), Article ID 5307697, Volume 2022 (2022)





A Decentralized Electronic Reporting Scheme with Privacy Protection Based on Proxy Signature and Blockchain

Huiying Zou, Xiaofan Liu , Wei Ren , and Tianqing Zhu
Research Article (8 pages), Article ID 5424395, Volume 2022 (2022)

A Hybrid Design of Linkable Ring Signature Scheme with Stealth Addresses

Weizhou Li, Zhiqiang Lin , and Qi Chen
Research Article (9 pages), Article ID 1417607, Volume 2022 (2022)

Blockchain-Based Proof of Retrievability Scheme









Yan Ren , Haipeng Guan , Qiuxia Zhao , and Zongxiang Yi 
Research Article (8 pages), Article ID 3186112, Volume 2022 (2022)

Contents





An Android Malicious Application Detection Method with Decision Mechanism in the Operating Environment of Blockchain

Xingyu Li , Zongqu Zhao , Yongli Tang , Jing Zhang , Chengyi Wu , and Ying Li 
Research Article (10 pages), Article ID 3111540, Volume 2022 (2022)






BCST-APTS: Blockchain and CP-ABE Empowered Data Supervision, Sharing, and Privacy Protection Scheme for Secure and Trusted Agricultural Product Traceability System

Guofeng Zhang , Xiao Chen , Bin Feng , Xuchao Guo , Xia Hao , Henggang Ren , Chunyan Dong , and Yanan Zhang 
Research Article (11 pages), Article ID 2958963, Volume 2022 (2022)

Attribute Set-Based Boolean Keyword Search over Encrypted Personal Health Records

Yu Lin , Lingling Xu , Wanhua Li , and Zhiwei Sun 
Research Article (13 pages), Article ID 9023141, Volume 2021 (2021)



KLPPS: A k -Anonymous Location Privacy Protection Scheme via Dummies and Stackelberg Game

Dongdong Yang , Baopeng Ye , Wenyin Zhang , Huiyu Zhou , and Xiaobin Qian 
Research Article (15 pages), Article ID 9635411, Volume 2021 (2021)






A Regulatable Data Privacy Protection Scheme for Energy Transactions Based on Consortium Blockchain

Yufeng Li , Yuling Chen , Tao Li , and Xiaojun Ren 
Research Article (11 pages), Article ID 4840253, Volume 2021 (2021)

Lodestone: An Efficient Byzantine Fault-Tolerant Protocol in Consortium Blockchains

Chen Shan  and Lei Fan 
Research Article (10 pages), Article ID 2507670, Volume 2021 (2021)

ForkDec: Accurate Detection for Selfish Mining Attacks

Zhaojie Wang , Qingzhe Lv , Zhaobo Lu , Yilei Wang , and Shengjie Yue 
Research Article (8 pages), Article ID 5959698, Volume 2021 (2021)

A Research on Traceability Technology of Agricultural Products Supply Chain Based on Blockchain and IPFS

Lejun Zhang , Weimin Zeng, Zilong Jin, Yansen Su, and Huiling Chen
Research Article (12 pages), Article ID 3298514, Volume 2021 (2021)

Research Article

A New Lattice-Based Blind Ring Signature for Completely Anonymous Blockchain Transaction Systems

Yi-Yang Xie, Xiu-Bo Chen , and Yi-Xian Yang

Information Security Center, State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Xiu-Bo Chen; flyover100@163.com

Received 24 February 2022; Revised 19 July 2022; Accepted 27 July 2022; Published 1 September 2022

Academic Editor: Anmin Fu

Copyright © 2022 Yi-Yang 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.

Blockchain technology has been widely applied in numerous industries with its decentralization, verifiability, distributivity, and immutability. However, the identity privacy security of blockchain users is facing serious threats because of the openness of traditional blockchain transaction information. Moreover, numerous traditional cryptographic algorithms used by blockchain transaction networks are difficult to attack quantum computing. In this paper, we propose a new lattice-based blind ring signature scheme in allusion to completely anonymous blockchain transaction systems. There into, the blind ring signature can implement the complete anonymity of user identity privacy in blockchain transactions. Meanwhile, lattice cryptography can availablely resist quantum computing attacks. Firstly, the proposed signature scheme has strong computational security based on the small integer solution (SIS) problem and a high sampling success rate by utilizing the techniques of rejection sampling from bimodal Gaussian distribution. Secondly, the proposed signature scheme can satisfy the correctness and security under the random oracle model, including anonymity, blindness, and one-more unforgeability. Thirdly, we construct a blockchain transaction system based on the proposed blind ring signature algorithm, which realizes the completely anonymous and antequantum computing security of the blockchain users' identity privacy. Finally, the performance evaluation results show that our proposed blind ring signature scheme has lower latency, smaller key size, and signature size than other similar schemes.

1. Introduction

Blockchain has gained much attention that is widely used in digital currency, medical, government services, and other applications, however, the security problems of blockchain have become increasingly prominent in recent years. As the data information needs to be jointly maintained by each node in the blockchain distributed network, it requires that the transaction information must be public, which will lead to the disclosure of personal identity privacy data. In many classical blockchain systems represented by Bitcoin [1], users utilize a string of numbers unrelated to their real identity information as the transaction address, which preliminarily realizes the anonymity of identity privacy. Unfortunately, because transactions in the Bitcoin network can be linked, attackers can discover users' real identity information by their blockchain addresses [2, 3]. Therefore, to realize the

veritable anonymity of the users' identity privacy, it is necessary to ensconce the relationship between users and their corresponding blockchain addresses.

The anonymity of identity information can be realized by ring signature and blind signature cryptography algorithms. Ring signature, developed from group signature [4], was first proposed by Rivest [5] in 2001. In the ring signature scheme, multiple users spontaneously constitute a ring and then randomly choose a member in the ring to sign the message. The signer uses his secret key and ring public keys of all members to generate a legal and valid ring signature. The ring signature prevents the exposure of the actual signer and invariably protects the signer's identity privacy. Another algorithm that can provide anonymity is the blind signature, which was first proposed by Chaum [6] in 1983. In the blind signature scheme, the signer can sign the message in case of unknowing the true content of the signature file. The sign

holder sends the blinded message to the signer for signature. The blind signature guarantees that signers hardly infer sign holders' real identity information through the blind message, which effectively protects sign holders' identity information privacy. In the blockchain transaction network, numerous anonymous transaction schemes are based on blind signature or ring signature [7–9]. However, the ring signature or blind signature can only guarantee the anonymity of a single user participating in the blockchain transaction, which cannot protect the identity privacy of both parties at the same time. To satisfy the complete anonymity of blockchain transaction users' identity privacy, it is significant to establish a blind ring signature scheme suitable for complete anonymous blockchain transactions. In 2005, Chan et al. [10] first proposed a blind ring signature algorithm, and since then, numerous blind ring signature schemes have been designed [11–13].

The security of traditional signature algorithms depends on integer decomposition, discrete logarithm and bilinear equivalent mathematical problems. Unfortunately, quantum computing can easily solve traditional difficult mathematical problems. Shor [14] proposed a quantum algorithm that lets RSA cryptography, elliptic curve cryptography, and cryptosystems based on bilinear pairings face serious security challenges. Grover [15] proposed a quantum search algorithm that could provide secondary acceleration for search problems, which seriously threatened the security of symmetric cryptography and the Hash function. Therefore, it is a key research work to find a cryptosystem that can resist quantum computing attacks.

Lattice cryptography is a kind of antequantum computing cryptography with strong security and high computational efficiency, which is widely used in digital signature algorithm design and blockchain transaction networks. Gentry et al. [16] first designed a signature algorithm with lattice trapdoor sampling, whose security depends on solving the SIS problem. Lyubashevsky [17] proposed a signature scheme without trapdoor sampling, which uses rejection sampling to greatly improve the sampling efficiency. Ducas et al. [18] designed a new signature algorithm with lattice rejection sampling, which further improves the sampling success rate through random sampling on bimodal Gaussian distribution. In 2018, Gao et al. [19] first proposed a postquantum blockchain system, which integrated a lattice-based signature algorithm. In 2022, Zou et al. [20] proposed a lattice-based proxy signature scheme for anonymous blockchain-enabled electronic reporting systems, which not only realized the anonymity of user identity but also solved the problem of misbehaviors untraceability on the blockchain. Moreover, Rückert [21] proposed the first blind lattice-based signature algorithm. Li et al. [22] proposed a new blind signature algorithm applied in blockchain anonymous transaction authentication on the lattice. In addition, Melchor et al. [23] designed the first ring signature algorithm based on lattice cryptography. To further improve the sampling success rate, Wang et al. [24] designed a new ring signature algorithm using

Lyubashevsky's rejection sampling signature [17]. In 2019, Le et al. [25] designed the first blind ring signature algorithm based on the SIS problem with rejection sampling. Moreover, numerous lattice-based blind signature and ring signature schemes have been proposed [26, 27].

In this paper, we design a new lattice-based blind ring signature algorithm in allusion to the completely anonymous blockchain transaction system. The constructed transaction system satisfies the requirements of the user's identity privacy protection and resistance to quantum attacks. There are three main contributions, which are as follows:

- (1) We propose a new lattice-based blind ring signature algorithm using the rejection sampling technology. Sampling on the bimodal Gaussian distribution can greatly improve the success rate. In addition, we give proof of correctness and security under the random oracle model, including anonymity, blindness, and one-more unforgeability.
- (2) We construct a completely anonymous blockchain transaction system based on the proposed blind ring signature and provide detailed processes of the anonymous transaction. The system satisfies the goal of blockchain users' identity privacy protection and antequantum computing security.
- (3) We evaluate the performance of the proposed signature algorithm with other similar literature schemes, including the sampling method, algorithm latency, the size of the signature, and secret and public keys. The evaluation results indicate that our proposed scheme has lower latency and smaller key and signature sizes than other similar schemes.

The organization of this paper is as follows: we present some lattice theories and the blind ring signature's definition and security model in Section 2. In Section 3, a new lattice-based blind ring signature is designed. In Section 4, we prove the security of our signature algorithm. We construct a completely anonymous blockchain transaction system based on the proposed blind ring signature in Section 5. The performance evaluation of signature algorithms is shown in Section 6. Finally, we provide a conclusion of the paper in Section 7.

2. Preliminaries

2.1. Some Related Theories of Lattice

Definition 1 (Lattice [28]). Given a matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ consists of a group of m -dimensional linearly independent vectors $\mathbf{b}_1, \dots, \mathbf{b}_2, \dots, \mathbf{b}_n$, where $m \geq n$. Define lattice Λ generated by \mathbf{B} as the set.

$$\Lambda(\mathbf{B}) = \{\mathbf{B}\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^n\}. \quad (1)$$

Given a prime number q , a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and $\mathbf{e} \in \mathbb{Z}_q^n$, define some q -ary lattices.

$$\begin{aligned}
\Lambda_q(\mathbf{A}) &= \{\mathbf{y} \in \mathbb{Z}^m \mid \mathbf{y} \in \mathbf{A}^T \mathbf{x} \bmod q, \mathbf{x} \in \mathbb{Z}^n\}, \\
\Lambda_q^\perp(\mathbf{A}) &= \{\mathbf{y} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{y} = \mathbf{0} \bmod q\}, \\
\Lambda_q^e(\mathbf{A}) &= \{\mathbf{y} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{y} = \mathbf{e} \bmod q\}.
\end{aligned} \tag{2}$$

Definition 2 (Discrete Gaussian Distribution [17]). Define $D_{\mathbf{v},\sigma}^m(\mathbf{z}) = \rho_{\mathbf{v},\sigma}^m(\mathbf{z})/\rho_{\mathbf{v},\sigma}^m(\mathbb{Z}^m)$ as a discrete Gaussian distribution, where $\rho_{\mathbf{v},\sigma}^m(\mathbf{z}) = (1/\sqrt{2\pi\sigma^2})^m e^{-\|\mathbf{z}-\mathbf{v}\|^2/(2\sigma^2)}$ and $\rho_{\mathbf{v},\sigma}^m(\mathbb{Z}^m) = \sum_{\mathbf{z} \in \mathbb{Z}^m} \rho_{\mathbf{v},\sigma}^m(\mathbf{z})$.

Definition 3 (SIS problem). Given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and parameters m, n, q, β , the SIS $_{q,n,m,\beta}$ problem is to find a nonzero integer vector $\mathbf{v} \in \mathbb{Z}_q^m$, such that $\mathbf{A}\mathbf{v} = \mathbf{0} \pmod{q}$ and $\|\mathbf{v}\| \leq \beta$.

Lemma 1 (see [17]). For any $\mathbf{v} \in \mathbb{R}^m$, $\sigma > 0$, $k > 1$, it satisfies the following:

$$\Pr\{\|\mathbf{z}\| > k\sigma\sqrt{m}; \mathbf{z} \leftarrow D_{\sigma}^m\} \leq k^m e^{(m/2)(1-k^2)}. \tag{3}$$

Lemma 2 (see [17]). For any $\mathbf{v} \in \mathbb{Z}^m$, $\sigma = \alpha\|\mathbf{v}\|$, $\alpha > 0$, it satisfies the following:

$$\Pr\left\{\frac{D_{\sigma}^m(\mathbf{z})}{D_{\mathbf{v},\sigma}^m(\mathbf{z})} < e^{(12/\alpha)+(1/(2\alpha^2))}; \mathbf{z} \leftarrow D_{\sigma}^m\right\} > 1 - 2^{-100}. \tag{4}$$

More specially, if $\alpha = 12$, $\sigma = 12\|\mathbf{v}\|$, then $(D_{\sigma}^m(\mathbf{z})/D_{\mathbf{v},\sigma}^m(\mathbf{z})) < e^{1+(1/288)}$ with a probability of at least $1 - 2^{-100}$.

Lemma 3 (Rejection Sampling [17]). Select a random vector $\mathbf{v} \in \mathbb{Z}^m$ and a real number $\sigma = \omega(t\sqrt{\log m})$, given a subset $V = \{\mathbf{v} \in \mathbb{Z}^m: \|\mathbf{v}\| < t\}$, and define on V a probability distribution $h: V \rightarrow \mathbb{R}$. Then, there exists a constant $M = O(1)$ such that the outputs of the following two algorithms **A** and **B** have a negligible statistical distance of $\Delta(\mathbf{A}, \mathbf{B}) = 2^{-\omega(\log m)}/M$:

Algorithm A: $\mathbf{v} \leftarrow h$, $\mathbf{z} \leftarrow D_{\mathbf{v},\sigma}^m$, output (\mathbf{z}, \mathbf{v}) with probability $\min(D_{\sigma}^m(\mathbf{z})/(MD_{\mathbf{v},\sigma}^m(\mathbf{z})), 1)$.

Algorithm B: $\mathbf{v} \leftarrow h$, $\mathbf{z} \leftarrow D_{\sigma}^m$, output (\mathbf{z}, \mathbf{v}) with probability $1/M$.

Moreover, the probability that the algorithm **A** outputs something is at least $(1 - 2^{-\omega(\log m)})/M$.

More specially, if $\sigma = at$ for any $\alpha > 0$, then $M = e^{(12/\alpha)+(1/(2\alpha^2))}$. The two algorithms **A** and **B** have a negligible statistical distance off $\Delta(\mathbf{A}, \mathbf{B}) = 2^{-100}/M$, and the probability that **A** outputs something is at least $(1 - 2^{-100})/M$.

2.2. Blind Ring Signature Model

2.2.1. System Model. The blind ring signature system model is composed of four parts called setup, key generation, signature, and verification [25]. The detailed steps are as follows:

Setup. Input a security parameter n and output public parameters PP.

Key generation. Generate public key pk and secret key sk for each member of the ring $R = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_l\}$ according to the input set of public parameters PP.

Signature. The user Y submits a message m and blinds it to μ before sending the message to the signer. Then, the ring R chooses a signer Σ_j , who takes the secret key sk_j . The signer Σ_j signs the message μ and generates a blinded signature Σ' . The user Y unblinds Σ' and gets the real signature Σ .

Verification. Output 1 or 0 according to the public parameters PP, message m , signature Σ , and ring public keys $PK = \{pk_i\}_{i \in [l]}$. The output of 1 means that the verification is passed, and 0 indicates that it is otherwise.

2.2.2. Security Model. The security model of the blind ring signature includes anonymity, blindness, and one-more unforgeability.

Anonymity: the anonymity property ensures that the user cannot know which member of the ring was the real signer participating in the blind ring signature protocol. For any polynomial-time adversary, the blind ring signature scheme satisfies the anonymity under full key exposure if his advantage in winning the following game with the challenger is negligible.

- (1) **Setup:** assume n to be the system security parameter. The challenger calls the setup algorithm in the blind ring signature scheme to generate the set of common parameters PP. Then, according to the common parameters PP, the challenger calls the key generation algorithm to generate a set of public and secret keys (PK, SK) for the ring $R = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_l\}$. The challenger sends the set of common parameters PP and public key PK to the adversary .
- (2) **Query:** the adversary submits a message m , a ring R , an index I , and the corresponding public key pk_i to the challenger . The challenger queries the corresponding secret key sk_i according to the index I and then calls the signature algorithm to generate a blinded signature Σ'_i on m for the adversary .
- (3) **Challenge:** the adversary submits a message m , a ring R , and two public keys $pk_{i_b} \in R$ to the challenger for the signature query, where $b \in \{0, 1\}$. The challenger chooses a random bit $b \in \{0, 1\}$. Then, it uses the secret key sk_{i_b} and calls the signature algorithm to generate a blinded signature Σ'_i on m and returns Σ'_i to the adversary .
- (4) **Guess:** the adversary outputs a bit b' as a guess of the random bit b . He wins the game if $b' = b$.

The advantage of the adversary in the above game is defined as follows:

$$\text{Adv}_{\text{BRS}}^{\text{anonymity}}(A) = \left| \Pr\{b' = b\} - \frac{1}{2} \right|. \tag{5}$$

Blindness: it is a basic attribute of the blind ring signature, i.e., all members in the ring cannot know any information about the message to be signed. In other words, the attacker cannot distinguish the original signature of which message a blind ring signature comes from. For any polynomial-time adversary, the blind ring signature scheme satisfies the statistical blindness if his advantage in winning the following game with the challenger is negligible.

- (1) **Setup:** assume n to be the system security parameter. The challenger calls the setup algorithm in the blind ring signature scheme to generate the set of common parameters PP. Then, according to the common parameters PP, the challenger calls the key generation algorithm to generate a set of public and secret keys (PK, SK) for the ring $R = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_l\}$. The challenger sends the set of common parameters PP and public key PK to the adversary.
- (2) **Challenge:** the adversary α chooses two different blinded messages μ_0 and μ_1 , a subring $R' \in R$, and its corresponding public keys PK to send it to the challenger. The challenger chooses a random bit $b \in \{0, 1\}$, then sets up a blind ring signature protocol taking μ_b and the ring R' as input. The adversary chooses a signer Σ_j in the ring R' to sign the hidden blinded message μ_b . Finally, the adversary obtains the unblinded signature $\Sigma_b \neq \perp$, otherwise, it restarts this game.
- (3) **Guess:** the adversary outputs a bit b' as a guess of the random bit b . He wins the game if $b' = b$.

The advantage of the adversary in the above game is defined as follows:

$$\text{Adv}_{\text{BRS}}^{\text{blindness}}(A) = \left| \Pr\{b' = b\} - \frac{1}{2} \right|. \quad (6)$$

One-more unforgeability: the one-more unforgeability property ensures that the attacker cannot successfully forge a new correct signature through multiple signature inquiries. For any polynomial-time adversary, the blind ring signature scheme satisfies the one-more unforgeability if his probability of winning the following game with the challenger is negligible.

- (1) **Setup:** assume n to be the system security parameter. The challenger calls the setup algorithm in the blind ring signature scheme to generate the set of common parameters PP. Then, according to the common parameters PP, the challenger calls the Key generation algorithm to generate a set of public and secret keys (PK, SK) for the ring $R = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_l\}$. The challenger sends the set of common parameters PP and public key PK to the adversary. The secret key SK cannot be disclosed.
- (2) **Query:** the adversary submits a message m , a ring R , and its corresponding public keys PK. Then, adaptively, it makes multiple hash queries and blind ring signature queries to the challenger. The challenger

must return the hash value $H(m)$ and signature value Σ of the corresponding message m to the adversary.

- (3) **Forge:** the adversary uses the result of multiple queries to forge Σ^* of the target message m^* . One-more unforgeability requires that the pair (m^*, Σ^*) has never passed the signature verification algorithm.

3. Proposed Blind Ring Signature Algorithm

Our proposed blind ring signature algorithm includes five parts: key generation, message blinded, signature, unblind, and verification.

Key generation: Assume n is a system security parameter. We generate the common parameter PP, which has been selected by the same methodology of Li's scheme [22]. The independent public and secret key pairs $(\mathbf{A}_i, \mathbf{S}_i)$ for each signer $\mathcal{S}_i, i \in [l]$ of the ring $R = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_l\}$ are generated using the method described in Ducas's scheme [18], where $\mathbf{A}_i \in \mathbb{Z}_{2q}^{n \times m}$, $\mathbf{S}_i \in \mathbb{Z}_{2q}^{m \times n}$, and satisfying $\mathbf{A}_i \mathbf{S}_i = q \mathbf{I}_n \pmod{2q}$.

Message blinded: the signer of the ring R first computes a commitment to the user Y . Then, the user Y hides the original message m by running the message blinded algorithm and outputting the blinded message μ . The detail is shown in algorithm 1.

Signature: the ring R chooses a signer Σ_j . Σ_j calls the signature algorithm after receiving the blinded message μ and then outputs the blinded signature Σ' . The detail is shown in algorithm 2.

Unblind: the user Y runs the unblind algorithm after receiving the blinded signature Σ' and then outputs the real blind ring signature Σ . The detail is shown in algorithm 3.

Verification: the verifier runs the verification algorithm after receiving the original message m and blind ring signature Σ . Then, he outputs 1 if the verification is passed. It is 0, otherwise. The detail is shown in algorithm 4.

4. Correctness and Security Proof

4.1. Correctness. For the generated blind ring signature $\Sigma = (\{\mathbf{e}_i\}_{i \in [l]}, \mathbf{c})$, $\{\mathbf{e}_i\}_{i \in [l]}$ are sampled from the distribution $D_{\sigma_3}^m$, and according to Lemma 1, $\|\mathbf{e}_i\| \leq \eta \sigma_3 \sqrt{m}$ is established with an overwhelming probability for all $i \in [l]$. Therefore, the correctness is to prove $\sum_{i \in [l]} \mathbf{A}_i \mathbf{e}_i + q \mathbf{c} = \mathbf{x} + \mathbf{w} \pmod{2q}$. The proof of the equation is as follows:

$$\begin{aligned} \sum_{i \in [l]} \mathbf{A}_i \mathbf{e}_i + q \mathbf{c} &= \sum_{i \in [l]} \mathbf{A}_i \mathbf{y}_i + \sum_{i \in [l]} \mathbf{A}_i \mathbf{z}_i + q \mathbf{c} \\ &= \sum_{i \in [l] \setminus \{j\}} \mathbf{A}_i \mathbf{z}_i + \mathbf{A}_j \mathbf{z}_j + \sum_{i \in [l]} \mathbf{A}_i \mathbf{y}_i + q \mathbf{c} \\ &= \sum_{i \in [l] \setminus \{j\}} \mathbf{A}_i \mathbf{r}_i + \mathbf{A}_j (\mathbf{r}_j + \mathbf{S}_j \mu) + \sum_{i \in [l]} \mathbf{A}_i \mathbf{y}_i + q \mathbf{c} \\ &= \sum_{i \in [l]} \mathbf{A}_i \mathbf{r}_i + \mathbf{A}_j \mathbf{S}_j \mu + \sum_{i \in [l]} \mathbf{A}_i \mathbf{y}_i + q \mathbf{c} \\ &= \mathbf{x} + \mathbf{w} + q (-1)^t \mathbf{I}_n \mathbf{c} + q \mathbf{c} = \mathbf{x} + \mathbf{w} \pmod{2q}. \end{aligned} \quad (7)$$

Input: system public parameters PP, original message m , public keys $\{\mathbf{A}_i\}_{i \in [l]}$ of the ring R .
Output: blinded message μ .
Step 1: choose a set of random vectors $\{\mathbf{r}_i\}_{i \in [l]}$ from the bimodal Gaussian distribution $D_{\sigma_2}^m$.
Step 2: compute the commitment $\mathbf{x} = \sum_{i \in [l]} \mathbf{A}_i \mathbf{r}_i \pmod{2q}$.
Step 3: choose a set of blind factors $\{\mathbf{y}_i\}_{i \in [l]}$ from the bimodal Gaussian distribution $D_{\sigma_3}^m$.
Step 4: compute $\mathbf{w} = \sum_{i \in [l]} \mathbf{A}_i \mathbf{y}_i \pmod{2q}$.
Step 5: compute $\mathbf{c} = H(\mathbf{x} + \mathbf{w} \pmod{2q}, m)$.
Step 6: choose a random bit $t \leftarrow \{0, 1\}^n$.
Step 7: compute $\mu = (-1)^t \mathbf{c}$.
Step 8: output the blinded message μ with probability $\min(D_{\sigma_1}^m(\mu) / (M_1 D_{\mathbf{c}, \sigma_1}^m(\mu)), 1)$.

ALGORITHM 1: Message blinded algorithm.

Input: system public parameters PP, blinded message μ , the secret key \mathbf{S}_j of the signer Σ_j .
Output: blinded signature $\Sigma' = \{\mathbf{z}_i\}_{i \in [l]}$.
Step 1: for all $i \in [l] \setminus \{j\}$: compute $\mathbf{z}_i = \mathbf{r}_i$; for j : compute $\mathbf{z}_j = \mathbf{r}_j + \mathbf{S}_j \mu$.
Step 2: output \mathbf{z}_j with probability $\min(D_{\sigma_2}^m(\mathbf{z}_j) / (M_2 D_{\mathbf{S}_j \mu, \sigma_2}^m(\mathbf{z}_j)), 1)$.
Step 3: output the blinded signature $\Sigma' = \{\mathbf{z}_i\}_{i \in [l]}$.

ALGORITHM 2: Signature algorithm.

Input: system public parameters PP, blinded signature $\Sigma' = \{\mathbf{z}_i\}_{i \in [l]}$.
Output: blind ring signature $\Sigma = (\{\mathbf{e}_i\}_{i \in [l]}, \mathbf{c})$.
Step 1: for all $i \in [l]$: compute $\mathbf{e}_i = \mathbf{y}_i + \mathbf{z}_i$.
Step 2: output \mathbf{e}_i with probability $\min(D_{\sigma_3}^m(\mathbf{e}_i) / (M_3 D_{\mathbf{y}_i, \sigma_3}^m(\mathbf{e}_i)), 1)$.
Step 3: output the real blind ring signature $\Sigma = (\{\mathbf{e}_i\}_{i \in [l]}, \mathbf{c})$.

ALGORITHM 3: Unblind algorithm.

Input: system public parameters PP, original message m , public keys $\{\mathbf{A}_i\}_{i \in [l]}$ of the ring R , blind ring signature $\Sigma = (\{\mathbf{e}_i\}_{i \in [l]}, \mathbf{c})$.
Output: 1 or 0.
Step 1: verify that $\|\mathbf{e}_i\| \leq \eta \sigma_3 \sqrt{m}$ for all $i \in [l]$.
Step 2: verify that $\mathbf{c} = H(\sum_{i \in [l]} \mathbf{A}_i \mathbf{e}_i + q\mathbf{c} \pmod{2q}, m)$.
Step 3: output 1 if the verification in steps 1 and 2 passed and 0 otherwise.

ALGORITHM 4: Verification algorithm.

4.2. Security Proof

4.2.1. Anonymity. The adversary submits a message m and two users $\mathcal{U}_{i_0}, \mathcal{U}_{i_1} \in R$ to the challenger for a signature query. The challenger randomly chooses a bit $b \in \{0, 1\}$ and calls the message blinded algorithm and signature algorithm to generate a blinded signature $\Sigma'_{i_b} = \{\mathbf{z}_1, \dots, \mathbf{z}_{i_b}, \dots, \mathbf{z}_l\}$ on m , where $\mathbf{z}_{i_b} = \mathbf{r}_{i_b} + \mathbf{S}_{i_b} \mu$, output probability $\min(D_{\sigma_2}^m(\mathbf{z}_{i_b}) / M_2 D_{\mathbf{S}_{i_b} \mu, \sigma_2}^m(\mathbf{z}_{i_b}), 1)$, and $\mathbf{z}_i = \mathbf{r}_i \leftarrow D_{\sigma_2}^m$ for all $i \in [l] \setminus \{i_b\}$. Then, the challenger returns Σ'_{i_b} to the adversary. Let two random variables X_0 and X_1 represent the blinded signatures generated by the user \mathcal{U}_{i_0} and \mathcal{U}_{i_1} .

Suppose that the adversary obtains the blinded signature $\Sigma'_{i_b} = \{\mathbf{z}_1, \dots, \mathbf{z}_{i_b}, \dots, \mathbf{z}_l\}$ by sampling each \mathbf{z}_i from $D_{\sigma_2}^m$ with probability $1/M_2$, let the random variable Y represent the blinded signature generated by this way. The statistical distance [28] between X_0 and Y satisfies $\Delta(X_0, Y) \leq 2^{-\omega(\log m)} / M_2$, and the statistical distance between X_1 and Y satisfies $\Delta(X_1, Y) \leq 2^{-\omega(\log m)} / M_2$. Therefore, we have the following:

$$\Delta(X_0, X_1) \leq \Delta(X_0, Y) + \Delta(X_1, Y) \leq \frac{2^{1-\omega(\log m)}}{M_2}. \quad (8)$$

The statistical distance between X_1 and X_1 is negligible. Therefore, the distribution of blinded signatures Σ_{i_0}' and Σ_{i_1}' is indistinguishable. The proposed scheme satisfies anonymity.

4.2.2. Blindness. The adversary submits two different blinded messages, μ_0 and μ_1 , and interacts with two different users \mathcal{U}_{i_0} and \mathcal{U}_{i_1} . The adversary and the challenger only choose one of the two users for establishing an interactive blind ring signature protocol. It should be noted that the adversary does not know the user's information who is interacting with him, i.e., we can only prove that the outputs, i.e., the two blind messages μ_0 and μ_1 , are indistinguishable, and the corresponding blind ring signature Σ_{i_0} and Σ_{i_1} are also indistinguishable, where $b \in \{0, 1\}$ and $\Sigma_{i_b} = \{\mathbf{e}_1, \dots, \mathbf{e}_{i_b}, \dots, \mathbf{e}_l\}$.

For two blinded messages, μ_0 and μ_1 , because of the construction $\mu = (-1)^t \mathbf{c}$ and the output probability $\min(D_{\sigma_1}^m(\mu)/(M_1 D_{c, \sigma_1}^m(\mu)), 1)$, we can get that μ_0 and μ_1 are sampled from the same distribution $D_{\sigma_1}^m$. Therefore, the statistical distance between μ_0 and μ_1 satisfies $\Delta(\mu_0, \mu_1) = 0$ and they are indistinguishable. For two blind ring signatures Σ_{i_0} and Σ_{i_1} , because $\mathbf{e}_i = \mathbf{y}_i + \mathbf{z}_i$ for all $i \in [l]$ and the output probability $\min(D_{\sigma_3}^m(\mathbf{e}_i)/(M_3 D_{\mathbf{y}_i, \sigma_3}^m(\mathbf{e}_i)), 1)$, we can get Σ_{i_0} and Σ_{i_1} are sampled from the same distribution $D_{\sigma_3}^m$. Therefore, the statistical distance between Σ_{i_0} and Σ_{i_1} satisfies $\Delta(\Sigma_{i_0}, \Sigma_{i_1}) = 0$, and they are indistinguishable. The proposed scheme satisfies blindness.

4.2.3. One-More Unforgeability

Theorem 1. *If an adversary α can successfully give the effective forgery, there will be existing a polynomial-time algorithm Φ that can solve the SIS $_{q,n,lm,\beta}$ problem with non-negligible probability.*

Proof. We will prove the one-more unforgeability of the scheme by the simulation game between challenger and adversary. The simulation game controlled by challenger is executed as follows:

Setup: challenger builds two initial empty lists, List 1 and List 2, respectively, to store the hash value $H(m)$ and signature value $\Sigma = (\{\mathbf{e}_i\}_{i \in [l]}, \mathbf{c})$ of message m . Then, adversary will make hash queries and signature queries to challenger.

Hash queries: *The adversary* sends a hash query for message m to challenger. Challenger checks List 1, where List 1 consists of the pair $(m, H(m))$. If the queried message m is in List 1, challenger sends the corresponding $H(m)$ to *adversary*. If not, challenger will compute a new $H(m)$, restore $(m, H(m))$ into List 1, and send it to *adversary*.

Signature queries: *The adversary* sends a signature query for message m to challenger. The challenger checks List 2, where List 2 consists of the pair $(m, \Sigma = (\{\mathbf{e}_i\}_{i \in [l]}, \mathbf{c}))$. If the queried message m is in List 2, challenger sends the corresponding signature value

$\Sigma = (\{\mathbf{e}_i\}_{i \in [l]}, \mathbf{c})$ to adversary. If not, challenger will generate a new signature, restore the new pair $(m, \Sigma = (\{\mathbf{e}_i\}_{i \in [l]}, \mathbf{c}))$ into List 2, and send it to adversary.

Forge: suppose \mathbf{c}_j is a result of a hash query made by the adversary. Then, we can get the following:

$$\begin{aligned} & H\left(\sum_{i \in [l]} \mathbf{A}_i \mathbf{e}_i^* + q\mathbf{c}_j \pmod{2q}, m^*\right) \\ &= H\left(\sum_{i \in [l]} \mathbf{A}_i \mathbf{e}_i' + q\mathbf{c}_j \pmod{2q}, m'\right). \end{aligned} \quad (9)$$

For two different blind ring signature pairs, $(m^*, \Sigma^* = (\{\mathbf{e}_i^*\}_{i \in [l]}, \mathbf{c}_j))$ and $(m', \Sigma' = (\{\mathbf{e}_i'\}_{i \in [l]}, \mathbf{c}_j))$. We can find a hash collision if there exists inequality in the input of the hash function H on both sides of the equal sign of equation (10). Therefore, we can derive that $\sum_{i \in [l]} \mathbf{A}_i \mathbf{e}_i^* + q\mathbf{c}_j = \sum_{i \in [l]} \mathbf{A}_i \mathbf{e}_i' + q\mathbf{c}_j \pmod{2q}$, $m^* = m'$ with an overwhelming probability. Further simplification can be obtained as $\sum_{i \in [l]} \mathbf{A}_i (\mathbf{e}_i^* - \mathbf{e}_i') = 0 \pmod{2q}$. Let $\mathbf{e}_i = \mathbf{e}_i^* - \mathbf{e}_i'$, and we have the following:

$$\sum_{i \in [l]} \mathbf{A}_i \mathbf{e}_i = [\mathbf{A}_1 | \mathbf{A}_2 | \dots | \mathbf{A}_l] (\mathbf{e}_1^T, \mathbf{e}_2^T, \dots, \mathbf{e}_l^T)^T. \quad (10)$$

Let $\mathbf{A} = [\mathbf{A}_1 | \mathbf{A}_2 | \dots | \mathbf{A}_l] \in \mathbb{Z}^{n \times lm}$ and $\mathbf{e} = (\mathbf{e}_1^T, \mathbf{e}_2^T, \dots, \mathbf{e}_l^T)^T \in \mathbb{Z}^{lm}$. Then, we have $\mathbf{A}\mathbf{e} = \mathbf{0} \pmod{2q}$. As the forgery of the adversary is valid, there exists at least a bit i such that $\mathbf{e}_i^* \neq \mathbf{e}_i'$ and $\mathbf{e}_i^* - \mathbf{e}_i' \neq 0 \pmod{q}$ with an overwhelming probability, i.e., we can get $\mathbf{e} \neq \mathbf{0} \pmod{q}$ with great probability. Finally, we say that we can successfully solve the SIS problem. The detailed proving process is as follows:

Suppose that \mathbf{c}_j is a result of a hash query made by the adversary, and we can get a new valid forgery $\Sigma' = (\{\mathbf{e}_i'\}_{i \in [l]}, \mathbf{c}_j')$ for message m^* and ring R^* . We have $\mathbf{c}_j' \neq \mathbf{c}_j$ and $\sum_{i \in [l]} \mathbf{A}_i \mathbf{e}_i^* + q\mathbf{c}_j = \sum_{i \in [l]} \mathbf{A}_i \mathbf{e}_i' + q\mathbf{c}_j'$ with a non-negligible probability according to the Forking lemma [29]. Let $\mathbf{e}_i = \mathbf{e}_i^* - \mathbf{e}_i'$, $\mathbf{A} = [\mathbf{A}_1 | \mathbf{A}_2 | \dots | \mathbf{A}_l]$ and $\mathbf{e} = (\mathbf{e}_1^T, \mathbf{e}_2^T, \dots, \mathbf{e}_l^T)^T$. We have $\mathbf{A}\mathbf{e} = q(\mathbf{c}_j' - \mathbf{c}_j) \pmod{2q}$. Because $\mathbf{c}_j' \neq \mathbf{c}_j$ and $q(\mathbf{c}_j' - \mathbf{c}_j) = 0 \pmod{q}$, we can derive $\mathbf{e} \neq \mathbf{0} \pmod{2q}$ and $\mathbf{A}\mathbf{e} = \mathbf{0} \pmod{q}$. In addition, as $\|\mathbf{e}_i^*\| = \|\mathbf{e}_i'\| \leq \eta\sigma_3\sqrt{m}$ for all $i \in [l]$, according to algorithm 4, we have $\|\mathbf{e}_i\| = \|\mathbf{e}_i^* - \mathbf{e}_i'\| \leq \|\mathbf{e}_i^*\| + \|\mathbf{e}_i'\| = 2\eta\sigma_3\sqrt{m}$. Then, it satisfies $\|\mathbf{e}\| = \sum_{i \in [l]} \|\mathbf{e}_i\| \leq 2l\eta\sigma_3\sqrt{m}$. Therefore, \mathbf{e} is a solution to

the SIS $_{q,n,lm,\beta}$ problem with $\beta = 2l\eta\sigma_3\sqrt{m}$, where $\mathbf{A}\mathbf{e} = \mathbf{0} \pmod{q}$ and $\mathbf{e} \neq \mathbf{0} \pmod{q}$. The proposed scheme satisfies the one-more unforgeability. \square

5. The Completely Anonymous Blockchain Transaction System

In this section, we construct a completely anonymous blockchain transaction system based on the proposed lattice-based blind ring signature algorithm. Assume a blockchain transaction is required between Alice and Bob, and stipulate

that Alice transfers accounts to Bob. The transaction between Alice and Bob is recorded in a ledger and packaged into the blockchain. The overall schematic diagram of the anonymous blockchain transaction system is shown in Figure 1. The detailed process mainly includes the following five steps:

Key generation: firstly, Alice constructs a ring R composed of multiple members and calls the key generation algorithm and then gets the public and secret key pair (pk_A, sk_A) of ring R , where $pk_A = (pk_1, \dots, pk_a, \dots, pk_n)$ is a set of ring public keys.

Transaction generation: Bob initiates a transaction request with Alice and generates a piece of transaction information m . Bob and Alice run the blind ring signature algorithm in Section 3. Then, Bob selects the blind factor and utilizes the ring public keys pk_A of Alice to blind the transaction information m to μ . Alice uses the secret key sk_A to generate a signature Σ' for blinded transaction information μ . Bob obtains the real blind ring signature Σ of the transaction information m using the unblind algorithm. Finally, Bob generates a new transaction Tx utilizing the ring public keys pk_A and the blind ring signature Σ of the transaction information m .

Transaction authentication: Bob broadcasts the transaction Tx to the blockchain network, and the miner nodes in the blockchain use the ring public keys pk_A of Alice to verify whether the blind ring signature Σ is correct. It indicates that the transaction is correct if the verification passes, and then, it encapsulates the transaction Tx in a new block. Otherwise, the transaction will be discarded.

Network-wide consensus. The miners broadcast communication through the consensus mechanism and agree to add a new block containing the transaction to the blockchain. Meanwhile, miners who create the new block will be rewarded by the system.

Transaction completion: after blockchain miners have successfully reached the network-wide consensus on the transaction, Bob can consume the transfer received from Alice under the above steps.

The proposed transaction system has the characteristic of complete anonymity that can hide the identity privacy information of both parties participating in a blockchain transaction. For the internal attackers involved in the transaction, based on the blind signature feature, as the transaction initiator performs blind processing on the transaction information, the internal attacker cannot associate any veritable identity of the initiator through the transaction information. Therefore, for the input of each transaction, the internal attacker cannot trace whether it was initiated by the same user. For the external attackers not involved in the transaction, based on the ring signature feature, as the signature of the transaction is verified through ring public keys rather than a unique public key, it is impossible to determine the specific public key associated with

the real signer. Therefore, for the output of any two transactions, the external attacker cannot link to the same transaction user. Moreover, the signature algorithm adopted in this system is based on the SIS problem, which cannot be available solved by existing quantum computing algorithms. Therefore, the system satisfies antequantum computing security.

6. Performance Evaluation

In this section, we make an evaluation on the performance of the proposed signature algorithm by comparing with other similar literature schemes, including signature and verification algorithm latency, sampling method, the size of the signature, and secret and public keys. Firstly, we give some parameter settings, and then, the comparison results will be presented through theoretical analysis and simulation experiments.

6.1. Parameters Setting. The relevant public parameters of our scheme are set as shown in Table 1, which are the same as in [17]. We select the security level $k = 128$ bits and corresponding challenge size $\kappa = 28$ as an example. Meanwhile, the computational complexity of the SIS problem is maintained by reasonably selecting the parameter n, m, q , which can guarantee the security of public key and secret key. Moreover, the correctness error of the reject sample will be at the most 2^{-100} , which requires that $\sigma_1 = 12\|v\| = 12\sqrt{\kappa}$ and $M_1 = e^{12\sqrt{\kappa}/\sigma_1 + \kappa/2\sigma_1^2} = e^{1+1/288} \approx 2.72$. Then, M_2 and M_3 will be derived by the same method.

6.2. Comparison with Other Similar Schemes. We carry out the simulation experiment of efficiency comparison by utilizing MATLAB R2021b in the environment of Windows 11 with Intel(R) Core(TM) i7-10510U CPU 1.80 GHz and 16 G RAM. Assume that the same parameters (n, m, q, l, k, κ) , set according to Table 1, are utilized in each of these schemes, the detailed keys and signature size comparison results are shown in Table 2. We choose the parameters $l = 10, q = 2^{27}, k = 128$ and $\kappa = 28$ for the simulation experiment. Then, we compute the public key size, secret key size, and signature size for the different security parameter n , such as 80, 112, 128, 192, 256, 512. The comparison results of the public key size, secret key size, and signature size are separately shown in Figures 2–4. It can be seen from the experimental results that the size of the signature, secret, and public keys of our proposed scheme are all smaller than others [25, 30]. Moreover, we generate the public and secret keys without trapdoor sampling, which improves sampling efficiency and saves more time for performance.

Next, the results of the signature and verification algorithm latency comparison are shown in Table 3. The signature algorithm latency of the blind ring signature scheme includes message blinded, signature, and unblind algorithm latency. Here, some notations,

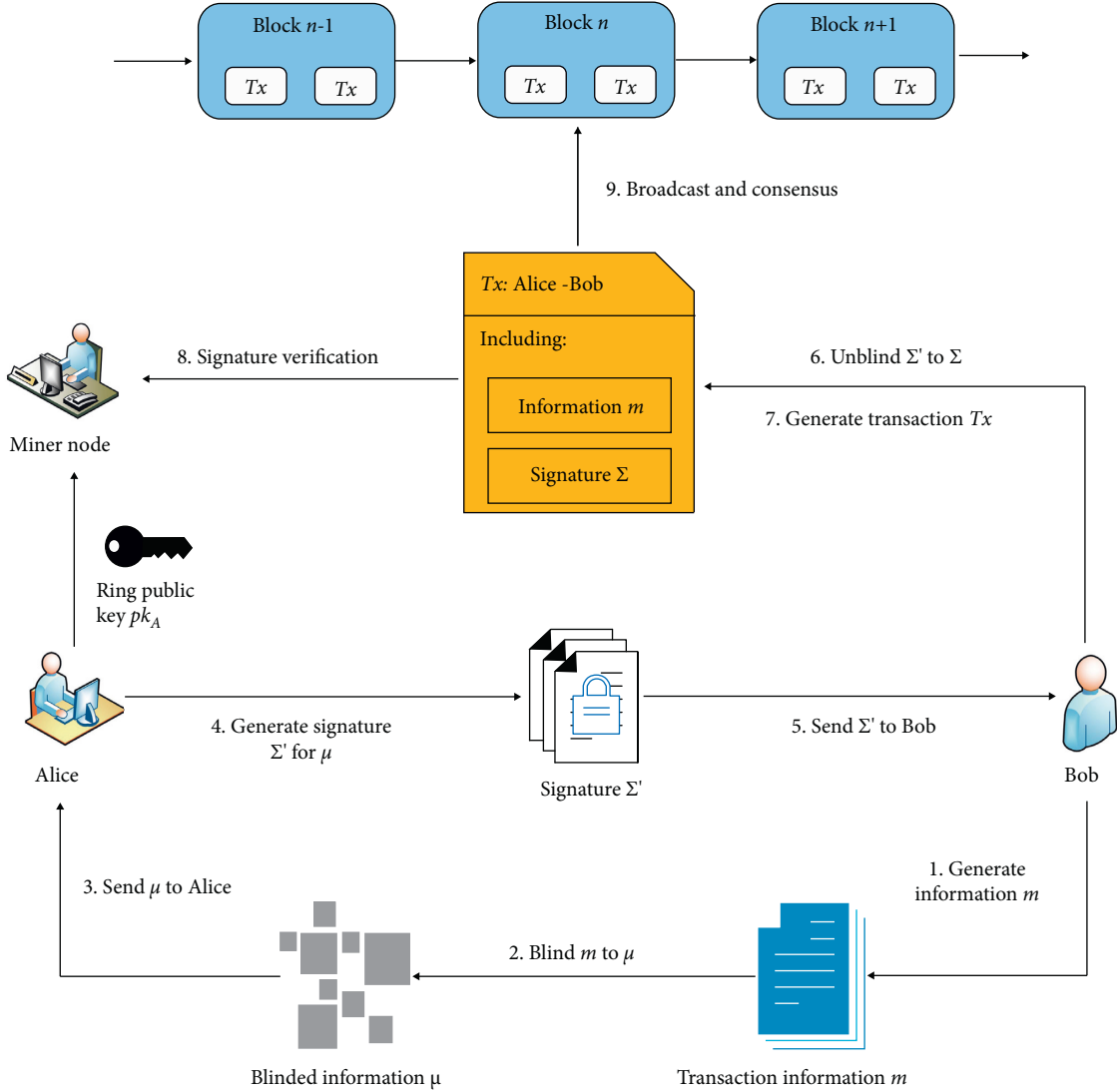


FIGURE 1: The completely anonymous blockchain transaction system.

TABLE 1: Public parameters setting.

Parameter PP	Definition	Example
N	Security parameter	512
l	Number of ring members	10
Q	$\text{poly}(n)$, prime	2^{27}
m	$m = n \log q$	13824
H	Hash function $H: \{0, 1\}^* \rightarrow \{c \in \{-1, 0, 1\}^k: \ c\ _1 \leq \kappa\}$	-
k and κ	In the hash function H and $2^\kappa \cdot C_k^\kappa \geq 2^{100}$	$k = 128, \kappa = 28$
η	[1.1, 1.3]	1.1
σ_1	$12\sqrt{\kappa}$	63
σ_2	$12\eta\sigma_1\sqrt{mk}$	2^{20}
σ_3	$12\eta\sigma_2\sqrt{m}$	2^{30}
$M_1 = M_2 = M_3$	$\exp(12\sqrt{\kappa}/\sigma_1 + \kappa/2\sigma_1^2)$	2.72
Secret key size	$lmn \log 2q$	236 MB
Public key size	$lmn \log 2q$	236 MB
Signature size	$lm \log(12\sigma_3) + \kappa$	0.55 MB

TABLE 2: Keys and signature size comparison.

Scheme	Public key size	Secret key size	Signature size	Sampling method
Wang et al. [30]	$6lmn \log q$	$l(6m)^2 \log q$	$6lm \log q + l$	Trapdoor sampling
Le et al. [25]	$6lmn \log q$	$6lmn \log q$	$6lm \log (12\sigma_3) + n + \kappa$	Trapdoor sampling
Our scheme	$lmn \log 2q$	$lmn \log 2q$	$lm \log (12\sigma_3) + \kappa$	Without trapdoor sampling

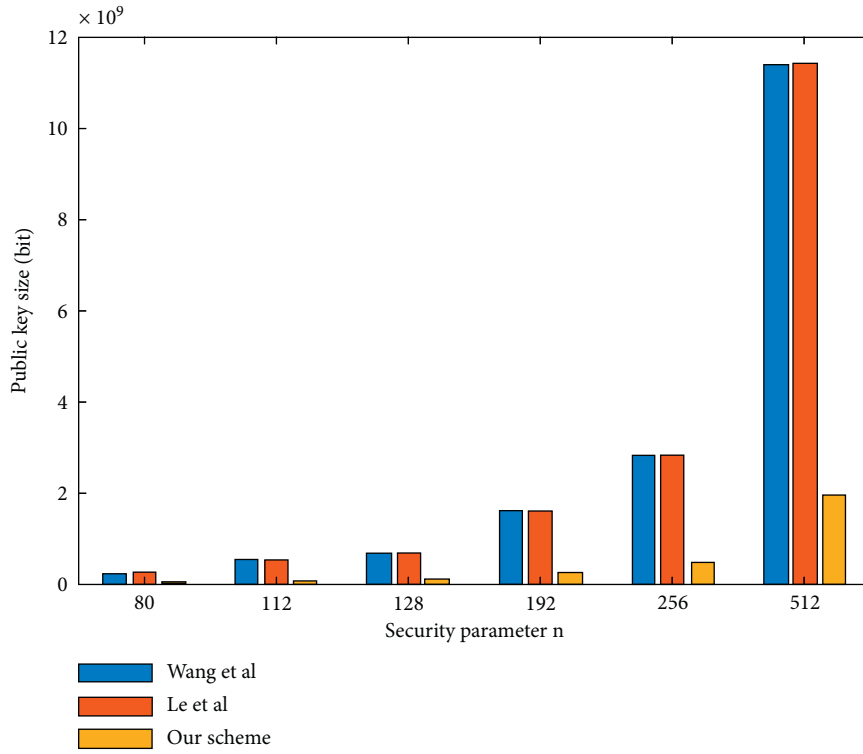


FIGURE 2: The comparison of public key size.

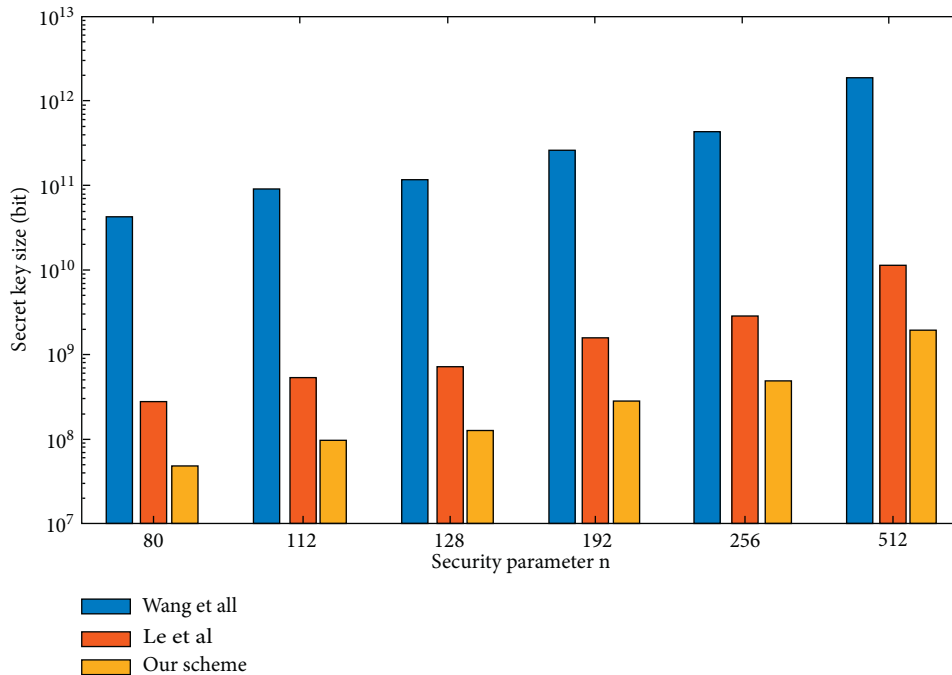


FIGURE 3: The comparison of secret key size.

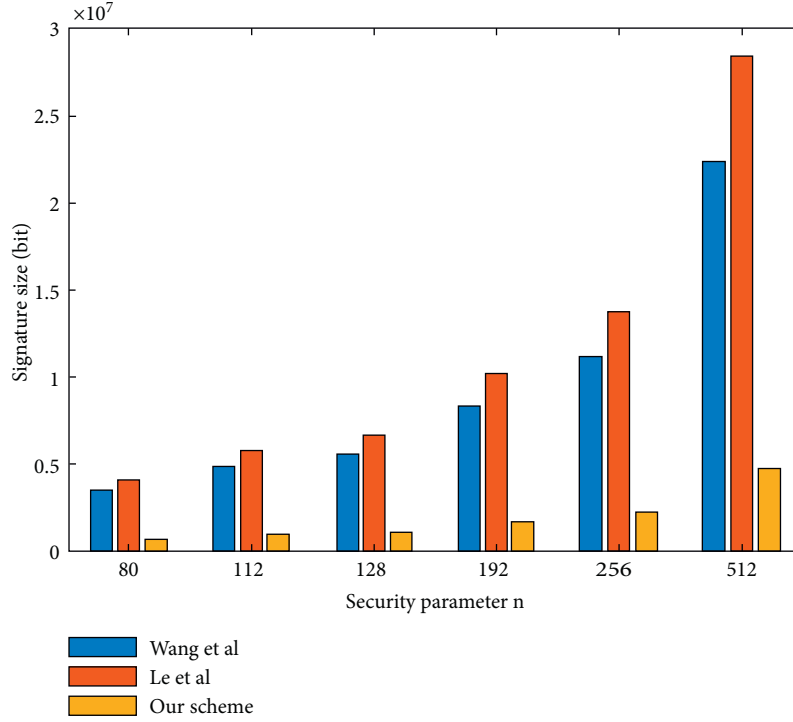


FIGURE 4: The comparison of signature size.

TABLE 3: Latency comparison.

Scheme	Signature algorithm latency	Verification algorithm latency
Le et al. [25]	$4(l+1)T_{Mul} + 2T_{Hash} + 3T_{RS} + T_{Com}$	$(l+1)T_{Mul} + T_{Hash} + T_{Com}$
Our scheme	$(2l+1)T_{Mul} + T_{Hash} + 2T_{RS}$	$(l+1)T_{Mul} + T_{Hash}$

such as T_{Mul} , T_{Hash} , T_{RS} , and T_{Com} , should be explained. The latency for multiplication is represented by T_{Mul} . The latency for the Hash operation is represented by T_{Hash} . The latency for rejection sampling operation is represented by T_{RS} . The latency for commitment function calculation is represented by T_{Com} . As can be seen from Table 3, our proposed blind ring signature scheme has lower signature and verification algorithm latency than the other similar scheme [25].

7. Conclusion

In this paper, we propose a new lattice-based blind ring signature scheme, which satisfies the correctness and security under the random oracle model, including anonymity, blindness, and one-more unforgeability. Meanwhile, the constructed blockchain transaction system based on our proposed blind ring signature satisfies the complete anonymity and antequantum computing security of users' identity privacy. Moreover, the proposed signature scheme has lower latency, smaller key, and signature sizes than other similar schemes.

However, our proposed scheme has some limitations. On the one hand, the proposed blind ring signature scheme relies on the difficult problem on the standard lattice, which leads to some disadvantages, such as large storage space of the key

matrix, low operation speed, and slow sampling rate, by comparing with structured lattice, such as ideal lattice. On the other hand, our constructed blockchain transaction system focuses on the implementation of user identity anonymity while ignoring the problem of double-spending attacks. In the future, firstly, we will study the linkable blind ring signature algorithm based on the ideal lattice to solve the limitations in the current work. Secondly, we will introduce the proposed blind ring signature algorithm into more specific blockchain application scenarios, such as medical blockchain and blockchain-enabled Internet of Things. Finally, we will study more cryptographic methods for blockchain data privacy protection, such as searchable encryption [32, 33], to improve blockchain privacy protection mechanisms.

Data Availability

The data and the code 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 to report regarding the present study.

Acknowledgments

This work was supported by the Fundamental Research Funds for Beijing Municipal Commission of Education, the Scientific Research Launch Funds of North China University of Technology, and Beijing Urban Governance Research Base of North China University of Technology.

References

- [1] S. Nakamoto, "Bitcoin: A Peer-To-Peer Electronic Cash System," *Decentralized Business Review*, vol. 21260, 2008.
- [2] M. Ober, S. Katzenbeisser, and K. Hamacher, "Structure and anonymity of the bitcoin transaction graph," *Future Internet*, vol. 5, no. 2, pp. 237–250, 2013.
- [3] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and Privacy in Social Networks*, pp. 197–223, Springer, New York, NY, USA, 2013.
- [4] D. Chaum and E. V. Heyst, "Group signatures," in *Workshop on the Theory and Application of Cryptographic Techniques*, pp. 257–265, Springer, Heidelberg, Germany, 1991.
- [5] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 552–565, Springer, Heidelberg, Germany, June 2001.
- [6] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology*, pp. 199–203, Springer, Boston, MA, USA, 1983.
- [7] X. Yi and K. Y. Lam, "A new blind ECDSA scheme for bitcoin transaction anonymity," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pp. 613–620, Association for Computing Machinery, New York, NY, USA, June 2019.
- [8] H. Yi, "A traceability method of biofuel production and utilization based on blockchain," *Fuel*, vol. 310, Article ID 122350, 2022.
- [9] Z. Wang and J. Fan, "Flexible threshold ring signature in chronological order for privacy protection in edge computing," *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 1253–1261, 2022.
- [10] T. K. Chan, K. Fung, J. K. Liu, and V. K. Wei, "Blind spontaneous anonymous group signatures for ad hoc groups," in *Proceedings of the European Workshop on Security in Ad-Hoc and Sensor Networks*, pp. 82–94, Springer, Heidelberg, Germany, July 2004.
- [11] Q. Wu, F. Zhang, W. Susilo, and Y. Mu, "An efficient static blind ring signature scheme," in *Proceedings of the International Conference on Information Security and Cryptology*, pp. 410–423, Springer, Heidelberg, Germany, May 2005.
- [12] H. Sun and Y. Ge, "New certificateless blind ring signature scheme," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 12, no. 1, pp. 778–783, 2014.
- [13] D. Hoang Duong, W. Susilo, and H. T. N. Tran, "A multivariate blind ring signature scheme," *The Computer Journal*, vol. 63, no. 8, pp. 1194–1202, 2020.
- [14] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, IEEE, Santa Fe, NM, USA, November 1994.
- [15] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pp. 212–219, Association for Computing Machinery, New York, NY, USA, July 1996.
- [16] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pp. 197–206, Association for Computing Machinery, New York, NY, USA, May 2008.
- [17] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 738–755, Springer, Heidelberg, Germany, June 2012.
- [18] L. Ducas, A. Durmus, T. Lepoint, and L. Vadim, "Lattice signatures and bimodal Gaussians," in *Proceedings of the Annual Cryptology Conference*, pp. 40–56, Springer, Heidelberg, Germany, July 2013.
- [19] Y. L. Gao, X. B. Chen, Y. L. Chen, Y. Sun, X. X. Niu, and Y. X. Yang, "A secure cryptocurrency scheme based on post-quantum blockchain," *IEEE Access*, vol. 6, Article ID 27205, 2018.
- [20] H. Zou, X. Liu, W. Ren, and Z. Tianqing, "A decentralized electronic reporting scheme with privacy protection based on proxy signature and blockchain," *Security and Communication Networks*, vol. 2022, Article ID 5424395, 8 pages, 2022.
- [21] M. Rückert, "Lattice-based blind signatures," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 413–430, Springer, Heidelberg, Germany, July 2010.
- [22] C. Li, Y. Tian, X. Chen, and J. Li, "An efficient anti-quantum lattice-based blind signature for blockchain-enabled systems," *Information Sciences*, vol. 546, pp. 253–264, 2021.
- [23] C. A. Melchor, S. Bettaieb, and X. Boyen, "Adapting Lyubashevsky's signature schemes to the ring signature setting," in *Proceedings of the International Conference on Cryptology in Africa*, pp. 1–25, Springer, Heidelberg, Germany, June 2013.
- [24] R. Zhao, S. Wang, and Y. Zhang, "Lattice-based ring signature scheme under the random oracle model," *International Journal of High Performance Computing and Networking*, vol. 11, no. 4, pp. 332–341, 2018.
- [25] H. Q. Le, D. H. Duong, and W. Susilo, "A blind ring signature based on the short integer solution problem," in *Proceedings of the International Workshop on Information Security Applications*, pp. 92–111, Springer, Cham, New York, NY, USA, June 2019.
- [26] G. Xu, Y. B. Cao, S. Y. Xu et al., "A novel post-quantum blind signature for log system in blockchain," *Computer Systems Science and Engineering*, vol. 41, no. 3, pp. 945–958, 2022.
- [27] C. H. Jiao and X. Y. Xiang, "Anti-quantum lattice-based ring signature scheme and applications in VANETs," *Entropy*, vol. 23, no. 10, p. 1364, 2021.
- [28] D. Micciancio and O. Regev, "Lattice-based cryptography," in *Post-Quantum Cryptography*, pp. 147–191, Springer, Heidelberg, Germany, 2009.
- [29] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 390–399, Association for Computing Machinery, New York, NY, USA, October 2006.
- [30] J. Wang and B. Sun, "Ring signature schemes from lattice basis delegation," in *Proceedings of the International Conference on Information and Communications Security*, pp. 15–28, Springer, Heidelberg, Germany, July 2011.

- [31] D. Micciancio and S. Goldwasser, "Complexity of lattice problems: a cryptographic perspective," *Springer Science & Business Media*, vol. 671, 2012.
- [32] G. Xu, S. Xu, Y. Cao, F. Yun, Y. Cui, Y. Yu et al., "PPSEB: a postquantum public-key searchable encryption scheme on blockchain for E-healthcare scenarios," *Security and Communication Networks*, vol. 2022, Article ID 3368819, 13 pages, 2022.
- [33] G. Xu, Y. Cao, S. Xu, X. Liu, X. B. Chen, and Y. Yu, "A Searchable Encryption Scheme Based on Lattice for Log Systems in Blockchain," *CMC-Computers Materials & Continua*, vol. 72, no. 3, pp. 5429–5441, 2022.

Research Article

A Privacy-Aware Electricity Consumption Data Collection Model Based on Group Blind Signature

Fengyin Li ¹, Xiao Li ¹, Peiyu Liu ², Xueqing Sun ¹, Siqi Yu ¹ and Junrong Ge ¹

¹School of Computer Science, Qufu Normal University, Rizhao 276826, China

²School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China

Correspondence should be addressed to Peiyu Liu; lpyu1960@126.com

Received 7 October 2021; Revised 14 November 2021; Accepted 27 April 2022; Published 19 May 2022

Academic Editor: Andrea Michienzi

Copyright © 2022 Fengyin Li 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 gives a new method for distributed data ledgering. The smart grid obtains efficient two-way data transmission and information control. It effectively monitors and regulates the grid by collecting real-time electricity consumption data of users. However, online data collection brings privacy leakage. To solve the problem of privacy leakage in the electricity data collection in the smart grid, a privacy-aware electricity data collection model is proposed. Firstly, we propose a new group blind signature scheme by introducing the blind feature into the identity-based encryption method. Secondly, by applying the proposed group blind signature scheme to the electricity data collection process, we propose a privacy-aware electricity data collection model. The proposed model ensures the conditional anonymity and traceability of user identity and the privacy protection and unforgeability of electricity consumption data.

1. Introduction

Blockchain technology originated from Satoshi Nakamoto's paper published in 2008. Blockchain, as a distributed shared ledger and database, in which records are copied and shared among its members, has the characteristics of decentralization, immutability, whole-process traces, openness, and transparency. Blockchain can store large decentralized data with better performance, availability, and scalability. Information leakage and low efficiency of blockchain are key issues that need to be addressed. A smart grid [1] is a new type of grid that combines traditional power grids with communication and information control technologies. It implements the two-way flow of the management information and power between the users and the power service provider. The architecture is shown in Figure 1. The smart grid is composed of four entities: power plant (PP), control center (CC), smart substation (SS), and smart meter (SM). There is a control center, several smart substations, and smart meters in a certain area, and the number of SSs is far less than SMs. Each SS is responsible for delivering power to users in a user area and collecting user electricity

consumption data. The SM submits the user's electricity consumption data to CC by SS. CC analyses users' electricity consumption data and arranges PP to generate power. The power arrives at SS in the form of high voltage through the high voltage transmission line. SS transforms high voltage power into low voltage power. Then, SS transmits power to a certain user area through the power distribution line.

Privacy protection issues are crucial in various systems, which are related to the reliability and security of the system. Chen et al. [2] proposed a visible routing approach PSSPR to achieve the source location privacy protection in WSNs. Li et al. [3] proposed a strong forward secure ring signature scheme based on RSA and introduced the ring signature into the privacy-aware PKI model, which achieves the privacy protection and user anonymity. Chen et al. [4] proposed a dynamic multi-key FHE scheme based on the LWE assumption in the public key setting. Otherwise, as a new biometric authentication technology, gait recognition [5–7] has attracted more and more researchers' attention in recent years. Some cloud computing-related works [8, 9] also help with privacy protection and data storage work greatly. In the smart grid, frequent information exchanges between SS and

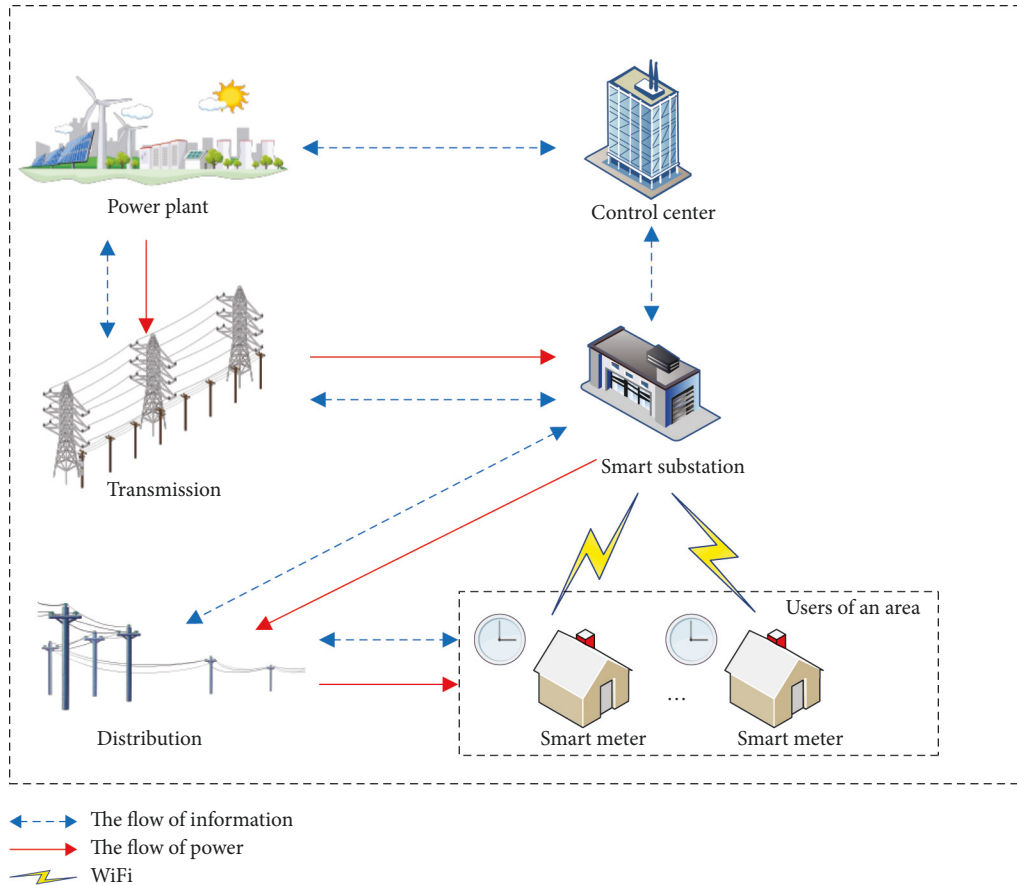


FIGURE 1: Architecture of smart grid.

SM bring privacy leakages [10, 11]. The adversary knows the user's daily schedule by eavesdropping on the electricity consumption data between SS and SM. Therefore, privacy protection in the smart grid receives more attention [12–15]. Zhao et al. [16] proposed a smart and practical privacy-preserving data aggregation scheme with smart pricing and packing method. Zhang et al. [17] proposed a blind signature-aided privacy-preserving power request scheme for a smart grid. The scheme protects the user's daily schedule. However, when the signature is invalid, CC cannot obtain the identity of the signer. The fine-grained requirements of the CC for electricity consumption data cannot be met. Some data aggregation schemes [18, 19] have been proposed in recent years. It is necessary to propose a method to implement user conditional anonymity and signer's traceability. In addition, CC obtains fine-grained electricity consumption data and verifies the integrity of the data.

Group blind signature technology provides a new way for us to achieve conditional anonymity and privacy protection for users in the smart grid. Group blind signature integrates the characteristics of group signature and blind signature at the same time. It allows the legal group member to anonymously generate signatures on behalf of the group. After the signing activity, the signer uses the group public key to verify the validity of the signature like others. However, he cannot know the signed time and who signed the signature. Due to the high anonymity and the traceability

of the group blind signature, more and more new practical schemes [20–22] have been proposed by domestic and foreign scholars. The group signature is applied in the electronic voting system [23], electronic cash system [24], intelligent transportation [25], and other fields to ensure system security. In recent years, the research of combining the group blind signature with quantum cryptography and lattice cryptography is also very popular [26–31].

In this study, we apply a new identity-based group blind signature to the privacy-aware electricity consumption data collection model. The model achieves user conditional anonymity and privacy protection. The contributions of this study are listed as follows:

- (1) By modifying the member-managing method, a new identity-based group blind signature scheme is proposed. The proposed group blind signature need not save the public keys of group members, which reduce the storage pressure of the system. The scheme effectively revokes the group members without updating the key of group manager and other group members.
- (2) Based on the proposed group blind signature scheme, a privacy-aware electricity consumption data collection model is proposed. Group blind signature assures the privacy of the electricity consumption data. In addition, we implement the user's

anonymous authentication and homomorphic verification tags, which ensure the privacy protection of user identity and the integrity of consumption data.

The organization of this study is as follows. Section 2 shows the preliminaries of this study. In Section 3, we propose a new group blind signature scheme and give its unforgeability proof. Then, we propose a privacy-aware electricity consumption data collection model based on group blind signature in Section 4. Section 5 shows the security and performance analysis of the data collection model. We conclude this study in Section 6.

2. Preliminaries

2.1. Elliptic Curve Discrete Logarithm Problem. The elliptic curve discrete logarithm problem (ECDLP) is that considering a point Q of prime order q on the elliptic curve E , if P is a possible point on E . It is difficult to find an $s \in Z_q^*$, which satisfied the equation $P = s \cdot Q$.

2.2. Group Blind Signature. A. Lysyanskaya and Z. Ramzan combined group signature and blind signature for the first time in 1998 to design the first group blind signature scheme-Lys98 scheme [32]. They used this scheme to construct an online and anonymous electronic cash system. The entities in the scheme usually contain the group manager, the group member, and the external user.

2.3. Homomorphic Tag. Homomorphism refers to mapping from one algebraic structure to another algebraic structure, and the anterior and posterior structure remains unchanged. The homomorphic tag is the tag with the property of homomorphism. Therefore, the tag of any two blocks of data can be computed from the sum of the tags of these two blocks. At the same time, when using the homomorphism tag to verify the integrity of the data, the verification can be completed only by verifying a part of the data block.

3. New Group Blind Signature Scheme

By introducing the blind feature into identity-based digital signature [33], this study proposes a new group blind signature scheme using the bilinear pairing mapping on the elliptic curve. The identity-based feature of the proposed scheme ensures that the signature system does not need to store the public key of group members. This feature reduces the storage overload of the system. In the new scheme, group members are effectively revoked without changing the key of the group manager and other group members. Security analysis indicates that the new scheme is reliable.

3.1. Group Blind Signature Scheme. Bilinear pairing is used to implement the identity-based group blind signature scheme. G_1 is an elliptic curve additive cyclic group whose order is a prime number q , and G_2 is a multiplicative cyclic group whose order is q . Meanwhile, a bilinear mapping is

$e: G_1 \times G_1 \longrightarrow G_2$. In the effective time, the discrete logarithm problem cannot be calculated whether in G_1 or G_2 .

3.1.1. System Initialization. The group manager selects generator $P \in G_1$ and three one-way hash functions: $H_1: \{0, 1\}^* \longrightarrow G_1$, $H_2: \{0, 1\}^* \longrightarrow Z_q^*$, and $H_3: G_1 \longrightarrow Z_q^*$. He chooses a random number $s \in Z_q^*$ as the private key and calculates $P_G = s \cdot P$ as his public key. Then, he initializes the group bulletin board $E = 1$ and the corresponding time T . The group manager releases system public parameters $\{G_1, G_2, P, P_G, H_1, H_2, H_3\}$ and announces the product of $H_2(ASID_i)$ on the group bulletin board, denoted as $E = \prod_i H_2(ASID_i)$.

3.1.2. Group Member Joining. A new member U_i joins this group. He first submits his real identity SID_i to the group manager. After the group manager verifies the validity of the identity, an anonymous identity code $ASID_i$ is generated. The member's public key Q_{ASID_i} and private key D_{ASID_i} are as follows:

$$Q_{ASID_i} = H_1(ASID_i), D_{ASID_i} = s \cdot Q_{ASID_i}. \quad (1)$$

The group manager saves $\langle SID_i, ASID_i \rangle$ in his database. Then, he sends the group member's private key D_{ASID_i} and anonymous identity code $ASID_i$ to U_i . At the same time, the group manager updates $E = H_2(ASID_i) \cdot E = H_2(ASID_i) \cdot \prod_i H_2(ASID_i)$. in the group bulletin board.

3.1.3. Group Member Revocation. The group manager updates the time T and E to revoke the member U_j . CC calculates $E = H_2(ASID_j)^{-1} \cdot E = H_2(ASID_j)^{-1} \cdot \prod_i H_2(ASID_i)$, where $j \in [0, i]$. At the same time, the group manager updates T to the present time. In this way, the group manager performs a multiplication operation to update E without changing the group public key and the group member key.

3.1.4. Group Blind Signature. For a received message, the group member signs it on behalf of the group. For instance, the signature steps of the group member $ASID_i$ are as follows:

- (1) A requester wants to acquire the signature of message m . He first chooses a random number $t_1 \in Z_q^*$ and calculates $m' = t_1 H_2(m)$. Then, he transmits m' to $ASID_i$.
- (2) After receiving m' , $ASID_i$ chooses a random number $k \in Z_q^*$ and calculates $R_1 = kP$, $S_1 = k^{-1}m'P$, and $S_2 = k^{-1}D_{ASID_i}$. Then, he sends the blind signature $\sigma t = (R_1, S_1, S_2, t)$ to requester, where t is the signature time.
- (3) requester chooses a random number $t_2 \in Z_q^*$ and calculates the signature $\sigma = (R, S, t)$ of message m as follows:

$$\begin{aligned}
R &= t_2 R_1 = t_2 k P, \\
S &= t_2^{-1} (t_1^{-1} S_1 + H_3(R) S_2), \\
&= t_2^{-1} \left(t_1^{-1} k^{-1} m P + H_3(R) k^{-1} D_{ASID_i} \right), \\
&= t_2^{-1} (t_1^{-1} k^{-1} t_1 H_2(m) P + H_3(R) k^{-1} D_{ASID_i}), \\
&= t_2^{-1} (k^{-1} (H_2(m) P + H_3(R) D_{ASID_i})).
\end{aligned} \tag{2}$$

3.1.5. Signature Verification. The validity verification of the signature $\sigma = (R, S, t)$ is divided into two steps. Firstly, the verifier selects the corresponding E based on the comparison between time t and T and verifies whether $H_2(ASID_i)$ is divisible by E . If $H_2(ASID_i)$ is not divisible, the signature is invalid. Otherwise, the signer is a member of the group. Then, the verifier uses the group public key P_G by comparing $e(R, S)$ with $e(P, P)^{H_2(m)} \cdot e(P_G, Q_{ASID_i})^{H_3(R)}$ to verify the validity of the signature. If the equation holds, σ is a validity signature. Otherwise, σ is invalid.

The verification process is as follows:

$$\begin{aligned}
e(R, S) &= e(t_2 k P, t_2^{-1} (k^{-1} (H_2(m) P + H_3(R) D_{ASID_i}))), \\
&= e(P, H_2(m) P + H_3(R) D_{ASID_i}), \\
&= e(P, P)^{H_2(m)} \cdot e(P, D_{ASID_i})^{H_3(R)}, \\
&= e(P, P)^{H_2(m)} \cdot e(P, s \cdot Q_{ASID_i})^{H_3(R)}, \\
&= e(P, P)^{H_2(m)} \cdot e(P_G, Q_{ASID_i})^{H_3(R)}.
\end{aligned} \tag{3}$$

3.2. Security Analysis. The group blind signature scheme proposed in this study satisfies unforgeability, anonymity, traceability, and revocability.

3.2.1. Unforgeability

Theorem 1. *If the ECDLP question is hard, under the existential unforgeability against chosen message attack (EU-CMA) model, the group blind signature scheme is existentially unforgeable.*

Proof. We assume that \mathcal{A} is an adversary authorized by a malicious user and able to forge group blind signatures. \mathcal{C} is a challenger who uses the adversary's ability to solve the ECDLP. However, this is contrary to the assumption of ECDLP, so the group blind signature scheme is secure. The group blind signature algorithm is modelled as a signing oracle, and the game is depicted as follows:

Setup: challenger \mathcal{C} performs the setup algorithm to generate system parameter and transmits it to \mathcal{A} . The system parameter includes $\{G_1, G_2, P, P_G, H_1, H_2, H_3\}$. \mathcal{C} randomly chooses an integer $i^* \in [1, l]$, where l denotes the maximum times of private key queries. Then, \mathcal{C} randomly chooses $s \in Z_q^*$ as the private key,

where the private key s' is equivalent to s . \mathcal{C} computes the public key $P_G = s \cdot P$.

Hash Queries: \mathcal{A} chooses the identity $ASID_i$ and sends to \mathcal{C} . \mathcal{C} calculates the hash value $Q_{ASID_i} = H_1(ASID_i)$ and sends it to \mathcal{A} .

Private Key Queries: \mathcal{A} makes the sign private key queries in this stage. \mathcal{C} maintains a list of legal signers. When \mathcal{A} queries signer's private key by sending the anonymous identity code $ASID_i$ to \mathcal{C} , \mathcal{C} checks the list of legal signers. When $i = i^*$, abort. When $i \neq i^*$, if $(i, ASID_i, Q_{ASID_i}, D_{ASID_i})$ exists, \mathcal{C} returns (Q_{ASID_i}, D_{ASID_i}) directly to \mathcal{A} . Otherwise, \mathcal{C} returns $Q_{ASID_i} = H_1(ASID_i)$ and $D_{ASID_i} = s' \cdot Q_{ASID_i}$ to \mathcal{A} and adds $(i, ASID_i, Q_{ASID_i}, D_{ASID_i})$ to the list of legal signers.

Sign Queries: adversary conducts signature queries at this stage. \mathcal{C} prepares a signature list to record all queries and responses. The list is empty at the beginning, and the format is (m, R_i, S_{1i}, S_{2i}) . \mathcal{A} selects the identity $ASID_i$ and message m , requesting the blind signature from \mathcal{C} . When \mathcal{A} queries the signature of $(m, ASID_i, Q_{ASID_i})$, if $i = i^*$, abort. Otherwise, \mathcal{C} randomly chooses $k \in Z_q^*$ and returns $\sigma'_i = (R_i, S_{1i}, S_{2i})$ to \mathcal{A} , where $R_i = kP$, $S_{1i} = k^{-1}mP$, and $S_{2i} = k^{-1}D_{ASID_i}$. Then, \mathcal{C} adds (m, R_i, S_{1i}, S_{2i}) to the signature list.

Outputs: \mathcal{A} finally outputs a valid forged signature $\sigma^* = (R_{i^*}, S_{1i^*}, S_{2i^*})$ of $ASID_{i^*}$ about the message m^* . In addition, \mathcal{A} cannot solve the ECDLP problem, so \mathcal{A} cannot get s' from $\sigma'_i = (R_i, S_{1i}, S_{2i})$. However, according to assumption \mathcal{A} gets the signature σ^* of message m^* . Therefore, \mathcal{C} obtains the solution s' of ECDLP according to the signature σ^* and the question previously queried.

Finally, \mathcal{A} solves the ECDLP assumption, but ECDLP is a difficult problem that cannot be calculated. Hence, under the difficulty assumption of ECDLP, the proposed group blind signature is existential unforgeability. \square

3.2.2. Anonymity. The correspondence $\langle SID_i, ASID_i \rangle$ between a group member's real identity SID_i and his anonymous identity code $ASID_i$ is only known by the group manager. Any other group members and external users cannot obtain it. The group member uses the anonymous identity to sign the message submitted by external users. No one obtains the real identity of the signer except the group manager, which implements the anonymity of the signer.

3.2.3. Traceability. The group member must submit his real identity SID_i to the group manager during the stage of group member joining. Then, he receives the anonymous identity code $ASID_i$ and the private key D_{ASID_i} . In this way, he becomes a legal group member and has the ability to sign messages. As long as the group member wants to correctly sign, he must use the anonymous identity code and private key distributed by the group manager. Therefore, the group manager has the ability to trace the real identity of the signer

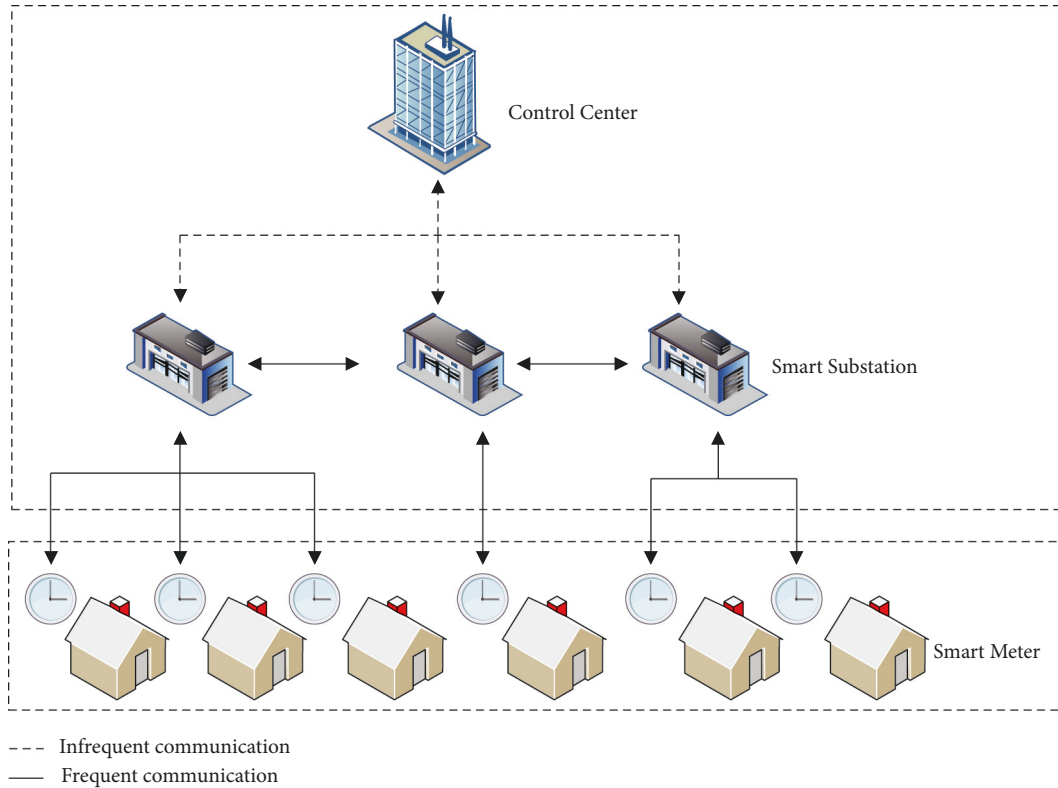


FIGURE 2: System model.

using $\langle SID_i, ASID_i \rangle$ stored in his database to achieve traceability.

3.2.4. Revocability. If a group member signs invalidly multiple times, he is identified as a malicious member. In this case, he revoked the group membership by the member revocation algorithm. Then, he loses the ability to sign on behalf of the group. Therefore, the proposed scheme has the revocability of group members.

4. A Privacy-Aware Electricity Consumption Data Collection Model Based on Group Blind Signature

By introducing the proposed group blind signature scheme into electricity consumption data collection, we propose a privacy-aware electricity consumption data collection model. The detail of the proposed model is as follows.

4.1. System Model. The system model in this study is shown in Figure 2, which involves three entities: control center (CC), smart substation (SS), and smart meter (SM). The working relationships and security requirements of the entities are as follows.

4.1.1. Control Center. CC generates system parameters, registers entities, verifies the electricity consumption data, and traces other entities conditionally. If the signature and

electricity data verification is invalid, CC traces the identity of the signer and user. CC exists in two forms, which are a fixed server located in the power plant and servers distributed in different places. CC needs to be highly credible.

4.1.2. Smart Substation. SS directly communicates with SM, verifies the user's identity, and generates the blind signature. SS does not know the user's real identity when he interacts with the user. SS needs to perform anonymous authentication on the user. SSs are fixed in certain places, generally.

4.1.3. Smart Meter. SM sends the regular electricity consumption data to CC. However, the electricity data may be tampered with within this process. Therefore, a reliable mechanism is needed to prevent the user's electricity data. SMs are installed in users' homes.

4.2. Adversary Model. The adversary model contains two main types of adversaries. One is the external adversary who is not in the data collection model. The other is the internal adversary who has the user's identity in the data collection model:

- (1) The external adversary obtains electricity consumption data by eavesdropping on the channel between SM and SS. The malicious forgery and replacement by the adversary threaten the integrity of the data.

TABLE 1: Description of notations in this study.

Notations	Descriptions
q, p	The large primes
G_1	The cyclic additive group
G_2	The cyclic multiplicative group
e	Bilinear pairing
P	A generator point
Z_q^*	Nonzero integers not larger than q
$H(\cdot)$	One-way hash function
s	Private key of group manager
P_G	Public key of group manager, where $P_G = s \cdot P$
T	The time of announcement E in the group bulletin board
E	The product of hashes of anonymous group members
SID_i	Real identity code of the group member
$ASID_i$	Anonymous identity code of the group member
D_{ASID_i}	Public key of group member, where $D_{ASID_i} = s \cdot Q_{ASID_i}$
Q_{ASID_i}	Private key of group member
m	The original message to be signed
mt	The blinded message
k	The random integer number
t_1, t_2	The blind factors
σ'	Blind signature
σ	Digital signature for m
n	Product of two large prime numbers, where $n = pq$
g	Primitive root of the modular n
a	Group public key, which is public key of RSA
b	Group private key, which is public key of RSA
x	Private key of group manager
y	Public key of group manager
$infor_i$	The information of user
gt_i	Encrypted value after user information has been hashed, where $gt_i = (H(infor_i))^x mo d n$
w_i	The random integer number
I_i	The pseudonym of user
λ	Security parameters of the electricity consumption data blocks generated by smart meters
stk	The private key of tag
ptk	The public key of tag
mx_j	The random value chosen by SM
u_j	The value needed to compute the tag is the same for each data block
tag_i	The value of tag for data block
M	The encrypted value of electricity data and the corresponding tag
TG	Multiplicative value of data block labels in a day
MG_j	The sum of the electricity data of jth dimension in a day
DG	Multiplications of bilinear pairing operation values for each data block
HS	The cumulative product of the hash value in a day

- (2) The internal adversary contains two types. One is the curious user who wants to acquire other users' electricity consumption data, but they do not tamper with any data. The other is the malicious user who tampers with his electricity consumption data.

4.3. Privacy-Aware Electricity Consumption Data Collection Model Based on Group Blind Signature. To protect the privacy of the user's identity and the electricity consumption data, an identity-based group blind signature scheme is used in the electricity consumption data collection model. CC and SSs form a group. SSs are group members, and CC is the group manager. We use the Schnorr identification protocol and the homomorphic verifiable tag mechanism to implement the anonymity of the user and the integrity verification of the electricity data. At the same time, the group blind signature mechanism ensures the traceability of the signer.

The notations used in this study are shown in Table 1.

In particular, the data collection model includes six stages: system initialization, anonymous identity authentication and data reporting, blind signature on reported electricity consumption data, data integrity verification and identity tracing, group member joining, and group member revocation. Figure 3 shows the framework of the data collection model.

4.3.1. System Initialization. In this stage, CC first generates system parameters. Then, the SS uses the real identity to apply for the group member private key and anonymous identity code. After CC verifies the identity of the SS, he distributes the anonymous identity code and the group member private key to the SS. CC saves the real identity and anonymous identity of SS in the database. SM also delivers its real information to CC and generates its own pseudonym.

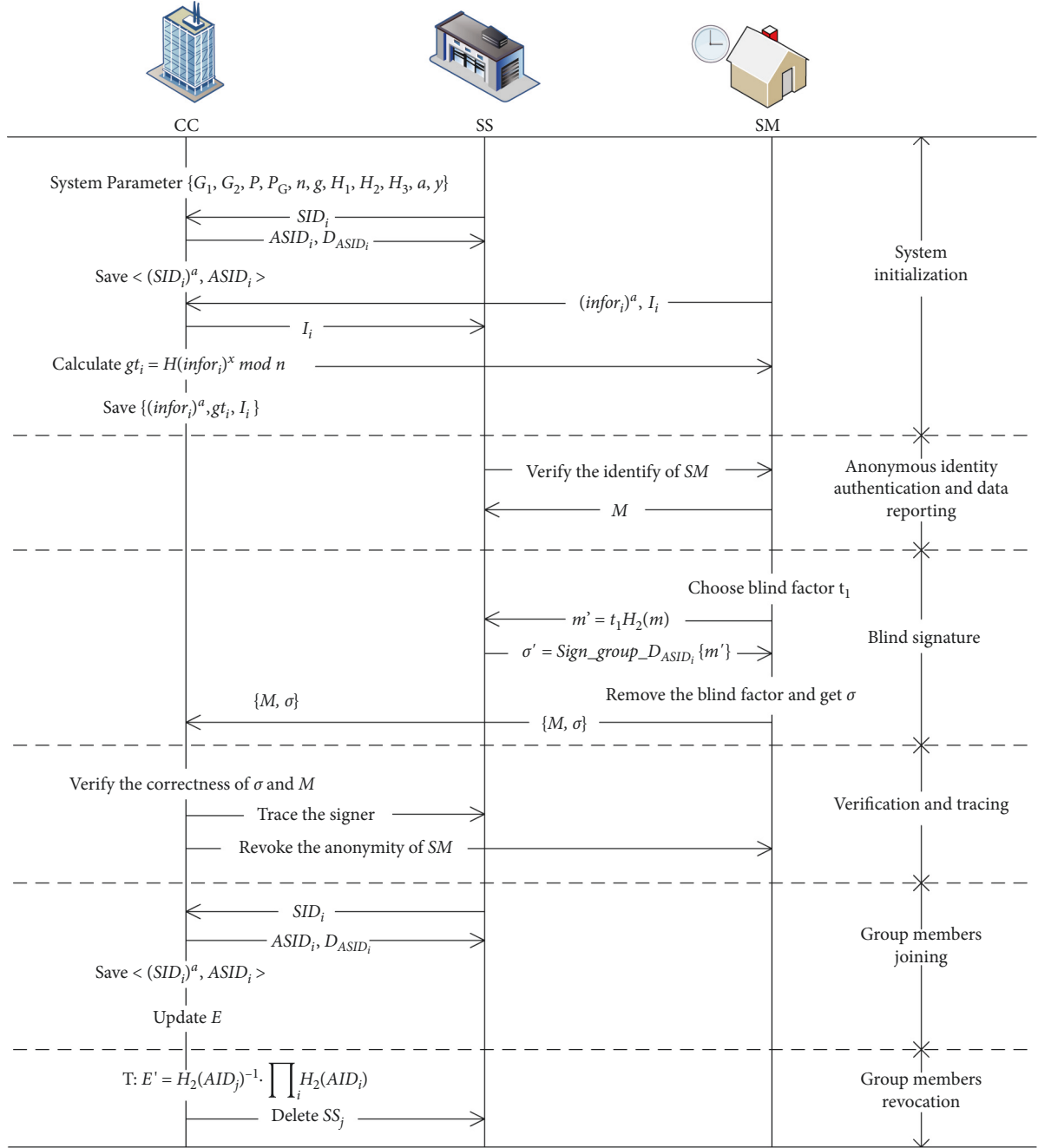


FIGURE 3: Framework of the data collection model.

CC saves the real identity and pseudonym of SM in the database. The data stored by CC, SS, and SM are, respectively, shown in Tables 2–4.

(1) Generating System Parameters.

- (i) CC computes $n = pq$, where p and q are two different large primes that meet $p|q - 1$.
- (ii) CC computes the group public key a and the private key b , where (a, b) satisfies the key pair property of RSA, namely $ab \equiv 1 \pmod{\phi(n)}$.
- (iii) CC chooses a random number $x \in [2, n - 2]$ and computes $y = g^x \bmod n$, where g is a

primitive root of the modular n . y and x are the public key and private key of the group manager, respectively.

- (iv) CC chooses generator $P \in G_1$ and three one-way hash functions: $H: \{0, 1\}^* \rightarrow \{0, 1\}^k$, $H_1: \{0, 1\}^* \rightarrow G_1$, $H_2: \{0, 1\}^* \rightarrow Z_q^*$, and $H_3: G_1 \rightarrow Z_q^*$.
- (v) CC chooses a random number $s \in Z_q^*$ as the system private key and computes $P_G = s \cdot P$ as the system public key. CC initializes the group bulletin board $E = 1$ and the corresponding time T . Then, CC releases system public

TABLE 2: Data stored by CC.

Definition	Symbol
Group public/private key	(a, b)
Group manager public/private key	(y, x)
System public/private key	(P_G, s)
The group bulletin board	E
The information of smart substation SS_i	$\langle (SID_i)^a, ASID_i \rangle (Q_{ASID_i}, D_{ASID_i})$
The information of smart meter SM_i	$(infor_i)^a H(infor_i)^x gt_i I_i$ (M, σ)

TABLE 3: Data stored by SS_i .

Definition	Symbol
Anonymous identity code	$ASID_i$
Public/private key	(Q_{ASID_i}, D_{ASID_i})
The pseudonym of SM_i	I_i
The blind signature	σ'

TABLE 4: Data stored by SM_i .

Definition	Symbol
The information of user	$infor_i$
The encrypted value of the information hash	gt_i
Random number	w_i
Pseudonym	I_i
Public/private tag key	(ptk, stk)
Electricity consumption data block and corresponding tag	(m, Tag)
The blind signature and signature of electricity consumption data	(σ, σ')

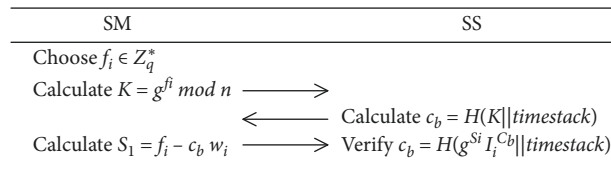


FIGURE 4: Anonymous identity authentication.

parameters $\{G_1, G_2, P, P_G, n, g, H_1, H_2, H_3, a, y\}$ and announces the time T and product of $H_2(ASID_i)$ on the group bulletin board, denoted as $E = \prod_i H_2(ASID_i)$.

(2) Registering Stage.

- (i) If SS_i wants to become a group member, he first submits real identity SID_i to CC. After CC verifies the validity of the identity, an anonymous identity code $ASID_i$ is generated. Then, CC calculates public key Q_{ASID_i} and private key D_{ASID_i} for SS_i as follows:

$$\begin{aligned} Q_{ASID_i} &= H_1(ASID_i), \\ D_{ASID_i} &= s \cdot Q_{ASID_i}. \end{aligned} \quad (4)$$

CC encrypts the real identity of the group member with the group public key a and saves $\langle (SID_i)^a, ASID_i \rangle$ in the database. Then, CC updates $E = H_2(ASID_i) \cdot E$.

- (ii) If a new user $User_i$ wants to participate in the smart grid. He first acquires $infor_i = (ID_i || \text{address} || \text{timestamp})$. Then, he encrypts his information $infor_i$ into $(infor_i)^a$ and sends it to

tag_1	tag_2	tag_3	tag_i	tag_{24}
m_1	m_2	m_3	m_i	m_{24}
m_{11}	m_{21}	m_{31}	m_{i1}	m_{241}
m_{12}	m_{22}	m_{32}	m_{i2}	m_{242}
m_{13}	m_{23}	m_{33}	m_{i3}	m_{243}
.....
m_{1j}	m_{2j}	m_{3j}	m_{ij}	m_{24j}
.....
m_{1l}	m_{3l}	m_{3l}	m_{il}	m_{24l}

FIGURE 5: Structure of electricity consumption data.

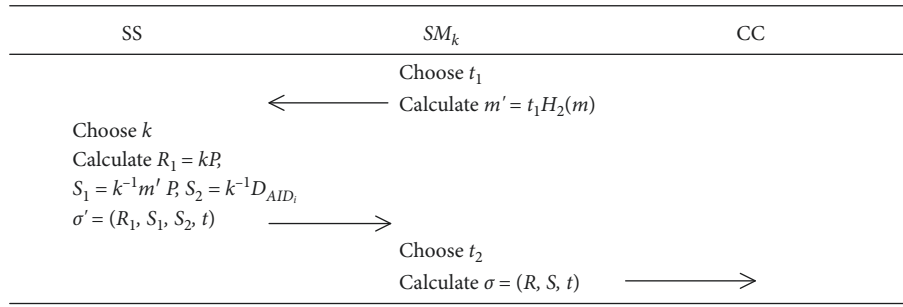


FIGURE 6: Generate the signature.

CC. CC stores $(infor_i)^a$ in his database and calculates $gt_i = (H(infor_i)^x) \bmod n$ sending it to $User_i$. The smart meters are distributed to users by CC. SM_i chooses a random number w_i to compute his pseudonym $I_i = g^{w_i} \bmod n$. SM_i sends I_i to CC.

4.3.2. Anonymous Identity Authentication and Data Reporting. In this stage, by the Schnorr identity authentication protocol, SM proves his legitimacy to SS under the condition of anonymity. Then, SM generates electricity consumption data blocks for a whole period. He calculates the data tag for each data block to ensure the integrity of data.

(1) *Anonymous Identity Authentication.* SS is not completely trusted in the model. When SM interacts with SS, the real identity of SM needs to be hidden. Therefore, the Schnorr identity authentication protocol is used to verify the legitimacy of SM. The authentication process is shown in Figure 4.

(2) *Data Reporting.* SS believes in the legitimacy of SM by anonymous identity authentication. Then, SM sends the encrypted electricity consumption data to SS. We take the example of $User_k$ encrypting and reporting electricity consumption data in one day. The whole day's data are m .

- (i) The data blocks generated in a day are restricted by the security parameter λ . We set the security

parameter λ to 24, and SM generates 24 data blocks in one day. The structure of data blocks generated in one day is shown in Figure 5. Each data block m_i represents one hour of electricity consumption data and has a corresponding tag tag_i . l -Dimensional attribute values are contained in each data block.

- (ii) SM_k randomly chooses the private tag key $stk \in Z_q^*$ and computes $ptk = gt_k^{stk} \bmod n$ as the public tag key.
- (iii) SM_k chooses l values $\{mx_1, mx_2, mx_3, \dots, mx_l\}$, randomly. Then, SM_k computes $u_j = gt_k^{mx_j} \bmod n$, where $j \in [1, l]$. SM_k calculates $tag_i = (H(MID \| i) \cdot \prod_{j=1}^l u_j^{m_{ij}})^{stk}$ for each data block m_i , where MID represents the data's summary and m_{ij} means the j th dimension attribute value of the i th data block. SM_k gets the tag set $Tag = \{tag_1, tag_2, tag_3, \dots, tag_i\}$, where the $i \in [1, 24/\lambda]$.
- (iv) SM_k computes $M = (m \| Tag)^a$ using the group public key a and calculates $H_2(m)$.

4.3.3. Blind Signature on Reported Electricity Consumption Data. In this stage, SM needs to get blind signature from SS. Then, SM reports the electricity consumption data and the signature to CC.

SS signs the electricity consumption data by the signature method provided in Section 3. SS sends the blind signature σ' to SM. SM removes the blind factor to get the signature σ . Then, SM sends σ and M to CC together. The signature generation process is shown in Figure 6.

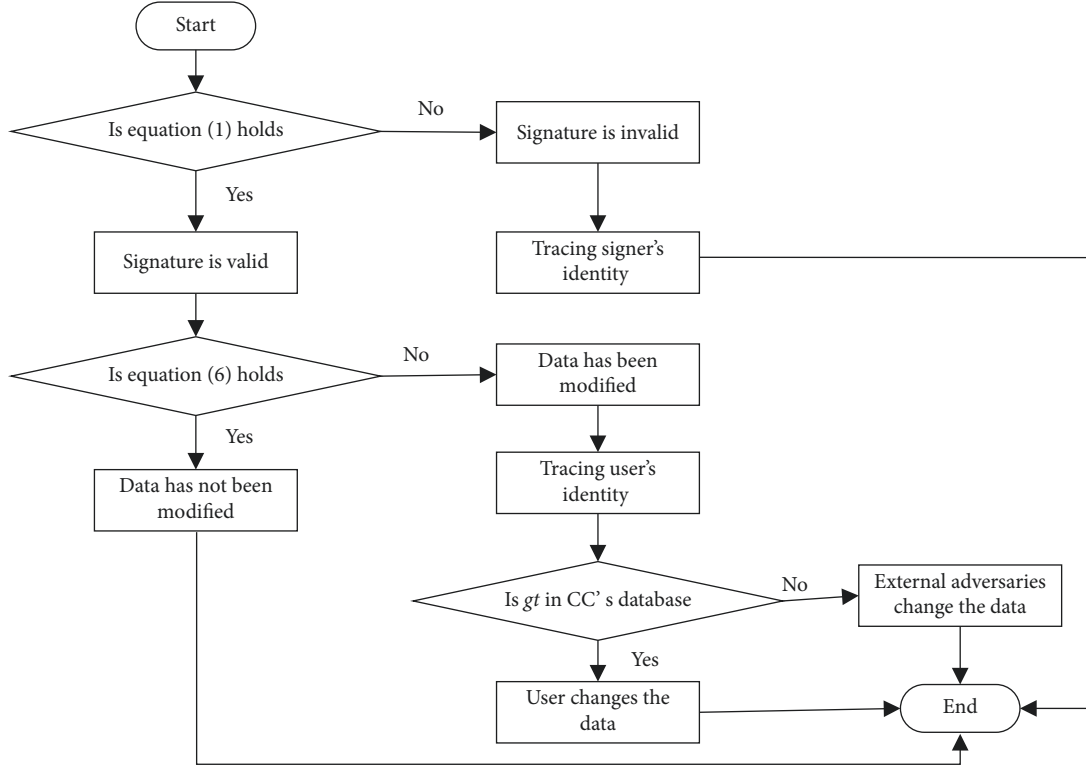


FIGURE 7: Flow chart of signature verification and identity tracing.

4.3.4. Data Integrity Verification and Identity Traceability. In this stage, CC verifies the validity of the signature and the integrity of data. Firstly, CC verifies the signature. If equation (1) holds, the signature is valid, it indicates that m has not been modified during the transmission process after being signed by SS. Otherwise, the signature is invalid. CC traces the signer's identity. Next, CC verifies the data integrity. CC uses Tag to check the integrity of the electricity consumption data. If equation (6) holds, the data are integral. Otherwise, the data have been modified. CC determines who tampered with the data. It is possible that the user or adversary has tampered with m before transmission. Therefore, CC obtains gt_k corresponding to M . CC compares gt_k with gt_i calculated using the user's real information stored in his database. If gt_k is in CC's database, it indicates that the user tampers with the data. Otherwise, the adversary tampers with the data. The flow chart of signature verification and identity tracing is shown in Figure 7.

(1) *The Verification of Signature and Data.* The verifier uses the group public key P_G to verify the validity of the signature $\sigma = (R, S, t)$. σ is the signature of M by the anonymous member $ASID_i$.

- (i) After receiving the M , CC gets the electricity consumption data m by decrypting the M . Then, CC computes $H_2(m)$ and the member's signature public key $Q_{ASID_i} = H_2(ASID_i)$.

- (ii) According to the comparison between time t and T , CC selects the corresponding E and verifies whether $H_2(ASID_i)$ is divisible by E . If $H_2(ASID_i)$ is not divisible, the signature is invalid. Otherwise, the signer is a member of the group. Then, CC uses equation 1 to verify the validity of the signature. If equation 1 holds, the signature is valid. Otherwise, the signature is invalid.

$$e(R, S) = e(P, P)^{H_2(m)} \cdot e(P_G, Q_{ASID_i})^{H_3(R)}. \quad (5)$$

- (iii) If the signature is valid, CC verifies the integrity of M . CC decrypts M to get Tag , m , u_j and calculates the following equations:

$$\begin{aligned} TG &= \prod_{i=1}^{24/\lambda} tag_i, \\ MG_j &= \sum_{i=1}^{24/\lambda} m_{ij}, \\ DG &= \prod_{j=1}^l e(u_j, ptk)^{MG_j}, \\ HS &= \prod_{i=1}^{24/\lambda} h(MID \parallel i). \end{aligned} \quad (6)$$

- (iv) CC verifies whether equation 6 holds every 24 hours:

$$DG \cdot e(HS, ptk) = e(TG, gt_k). \quad (7)$$

- (v) If equation 6 holds, the data have not been modified by the user or the adversary before transmission. Otherwise, the data have been modified.

(2) *Tracing the Signer and the User.*

- (i) If equation (1) does not hold, CC traces the signer's identity. The $\langle (SID_i)^a, ASID_i \rangle$ is saved in CC's database during the phase of group members joining. Therefore, CC uses the group private key b to decrypt $(SID_i)^a$ to obtain SID_i .
- (ii) If equation (1) holds and equation (6) does not hold, CC traces the user's identity to know who modified the data. CC uses the group private key b to decrypt $(infor_i)^a$ stored in his database to obtain $infor_i$. Then, CC calculates gt_i with the decrypted information one by one.

$$gt_i = H(infor_i)^x \bmod n = H(ID_i || \text{address} || \text{timestack})^x \bmod n. \quad (8)$$

Furthermore, CC compares gt_i with gt_k , which is corresponding to M to ensure the user's identity.

4.3.5. Joining of Group Members. A new member SS_i joins the group. SS_i first sends his real identity SID_i to CC through a reliable channel. CC generates an anonymous identity code $ASID_i$, the public key Q_{ASID_i} and the private key D_{ASID_i} for the new member SS_i . Then, CC updates $E = H_2(ASID_i) \cdot E = H_2(ASID_i) \cdot \prod_i H_2(ASID_i)$ in the group bulletin board.

4.3.6. Revocation of Group Members. CC updates the time T and E published on the group bulletin board to revoke the member SS_j . CC calculates the corresponding $E = H_2(ASID_j)^{-1} \cdot \prod_i H_2(ASID_i)$ at time T , where $j \in [0, i]$.

5. Security and Performance Analysis

The security and performance analysis section shows that the proposed data collection model is secure and reliable.

5.1. Security Analysis. The security of the model is mainly based on difficult problems, such as discrete logarithm problem, elliptic curve discrete logarithm problem, and integer decomposition problem. The following shows that the proposed model has the characteristics of privacy protection, anonymity, unforgeability, and traceability.

5.1.1. Privacy Protection

Theorem 2. *Due to the difficulty of the integer decomposition problem, the adversary cannot obtain the user's electricity consumption data.*

Proof. Adversary steals m when the user reports data and obtains the blind signature stage. However, the user's

electricity consumption data m are encrypted into M by the RSA encryption method. $M = (m || \text{Tag})^a$ can be decrypted only by the group private key b . In the data collection model, only CC has the group private key b . If the adversary wants to get m , he must obtain the private key b . The possible method is that the adversary solves the factor decomposition problem and decomposes n into correct p and q . Then, the adversary obtains the group private key b . However, the factor decomposition problem cannot be solved. The privacy protection of user electricity consumption data is implemented in our proposed model. \square

5.1.2. Anonymity. Anonymity includes the anonymity of the real identity of the SS and the real identity of the user who installed the SM.

(1) *Group Member Anonymity.* Only CC knows the correspondence $\langle (SID_i)^a, ASID_i \rangle$ between the anonymous identity and the real identity of SS. In the blind signature generation stage, the SS uses the anonymous identity to sign. Therefore, CC knows the real identity of the signer by a signature.

(2) *User Identity Anonymity.*

Theorem 3. *Because the discrete logarithm problem is difficult, \mathcal{A} cannot obtain the identity of the user by the decrypted electricity consumption data m and the corresponding tag Tag from the CC's database.*

Proof. The user's identity information $infor_i$ in CC's database is encrypted to $(infor_i)^a$. The RSA encryption is secure, and the adversary cannot calculate the group private key. Therefore, the adversary cannot obtain the user's identity information by decryption. If the adversary wants to get the user's identity, he calculates gt_k from the tag $tag_i = (H(MID || i) \cdot \prod_{j=1}^l u_j^{m_{ij}})^{stk} = (H(MID || i) \cdot \prod_{j=1}^l gt_k^{m_{ij}})^{stk}$. Then, he compares gt_k with $H(infor_i)^x \bmod n$ to determine the user's identity. However, the discrete logarithm problem is difficult, and the adversary cannot calculate gt_k from tag_i . The proposed model guarantees the anonymity of the user's identity information. \square

5.1.3. Unforgeability. Unforgeability includes the unforgeability of the group blind signature and the unforgeability of the user electricity consumption data.

(1) *Unforgeability of Group Blind Signature.* According to Theorem 1, we know whether the group blind signature is unforgeable.

(2) *Unforgeability of Electricity Consumption Data.* The adversary cannot forge the electricity consumption data. We use the homomorphic verifiable tag mechanism to verify the integrity of data. By judging whether equation (6) holds, we know whether the user's electricity consumption data have been forged or not. The detail is as follows:

$$\begin{aligned}
\text{Left} &= DG \cdot e(HS, pt_k) = \prod_{j=1}^l e(u_j, pt_k)^{MG_j} \cdot e(HS, pt_k), \\
\text{Right} &= e(TG, gt_k) = e\left(\prod_{i=1}^{24/\lambda} \text{tag}_i, gt_k\right), \\
&= e\left(\left(\prod_{i=1}^{24/\lambda} (H(MID\|i) \cdot \prod_{j=1}^l u_j^{m_{ij}})\right)^{stk}, gt_k\right), \\
&= e\left(\left(\prod_{i=1}^{24/\lambda} (H(MID|i))\right)^{stk}, gt_k\right) \cdot e\left(\prod_{j=1}^l \prod_{i=1}^{24/\lambda} u_j^{m_{ij}^{stk}}, gt_k\right), \\
&= e(HS, gt_k^{stk}) \cdot e\left(\prod_{j=1}^l \sum_{i=1}^{24/\lambda} m_{ij}, gt_k^{stk}\right), \\
&= e(HS, gt_k^{stk}) \cdot e\left(\prod_{j=1}^l u_j, gt_k^{stk}\right)^{\sum_{i=1}^{24/\lambda} m_{ij}}, \\
\prod_{j=1}^l e(u_j, pt_k)^{MG_j} &= e\left(\prod_{j=1}^l \sum_{i=1}^{24/\lambda} m_{ij}, gt_k^{stk}\right), \\
&= \prod_{j=1}^l e\left(u_j, gt_k^{stk}\right)^{\sum_{i=1}^{24/\lambda} m_{ij}}, \\
&= \prod_{j=1}^l e(u_j, gt_k^{stk})^{\sum_{i=1}^{24/\lambda} m_{ij}}, \\
&= e\left(\prod_{j=1}^l u_j, gt_k^{stk}\right)^{\sum_{i=1}^{24/\lambda} m_{ij}},
\end{aligned} \tag{9}$$

Left = Right.

Therefore, we know the integrity of the user's electricity consumption data by equation (6). The proposed model guarantees the unforgeability of user electricity consumption data.

5.1.4. Traceability. As shown in Section 4.3.4, CC traces the identity of the malicious signer and user under certain conditions.

If equation (1) does not hold, CC traces the identity of the signer. CC decrypts the $(SID_i)^a$ corresponding to the signer's anonymous identity code $ASID_i$ stored in his database. $(SID_i)^a = (SID_i)^{ab} = SID_i$. Then, CC obtains the signer's real identity SID_i . If equation (6) does not hold, CC

traces the user's identity. CC gets the user's registration identity information $(infor_i)^a$, which is stored in his database. CC decrypts $(infor_i)^a$ with the group private key b to obtain $infor_i$. Then, CC calculates $gt_i = H(infor_i)^x \bmod n$ of the $infor_i$ one by one. CC compares gt_i with the gt_k , which is corresponding to the M to ensure the user's identity. If gt_i is equal to gt_k , $infor_i$ is the user's real identity.

Therefore, the proposed model guarantees the traceability of the signer's identity and the user's identity.

5.2. Performance Analysis. In the performance analysis section, we analyse the calculation cost of the electricity data

collection model in four stages, including the system initialization stage, the user authentication stage, the blind signature stage, and the verification stage.

We assume to have α smart meters and β smart substations, where B stands for bilinear pairing operation, H stands for hash operation, M stands for modular multiplication operation, L stands for modular exponentiation operation, A stands for the elliptic curve addition operation, N represents the exponential operation under the multiplication group, and W stands for the elliptic curve multiplication operation. In the system initialization stage, CC computes y , gt_i , I_i , $H(infor_i)$, Q_{ASID_i} , P_G , and D_{ASID_i} . Therefore, $L = 1 + 2\alpha$, $H = \alpha + \beta$, and $W = 1 + \beta$. In the user authentication stage, the SM computes K and S_i and the SS computes c_b , so $M = \alpha$, $L = 3\beta$, and $H = 2\alpha$. Moreover, when data reporting, one SM within a day generates ptk , u_j , tag_i , $H_2(m)$, and $H(MID||i)$. Hence, the computational cost is $M = 24/\lambda$, $L = 1 + l + 24/\lambda$, and $H = 1 + 24/\lambda$. In the blind signature stage, the calculation cost of SM acquiring signatures in a day is $W = 7\alpha$, $A = \alpha$, and $H = \alpha$. In the verification stage, the computational cost of CC verifies that signatures in a day are $B = 3\alpha$, $H = 2\alpha$, and $N = 2\alpha$. The calculation cost of verifying the data within a day for one SM is $B = l + 2$, $M = 24/\lambda$, $H = 24/\lambda$, and $L = 24/\lambda$.

6. Conclusion

This study proposes a new identity-based group blind signature scheme and applies this signature scheme to the collection of user electricity consumption data in the smart grid. Then, we obtain a privacy-aware electricity consumption data collection model based on group blind signature. The model implements the conditional anonymity of user identity information and the privacy protection of consumption data in the process of collecting electricity data. In addition, when reporting electricity consumption data, the smart meter adds a tag to the data block generated every hour through the homomorphic tag mechanism. The user's electricity consumption data for a whole day correspond to a tag set. The existence of the tag ensures the integrity and verifiability of the electricity consumption data. The security and performance analysis proves that the data collection model has privacy protection, anonymity, unforgeability, and traceability. In future work, we consider combining blockchain technology with the proposed signature scheme in the smart grid scenario to protect the privacy of the user's electricity consumption data and identity information.

Data Availability

There are no data included in this study.

Conflicts of Interest

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

Acknowledgments

The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have greatly

improved the quality of the study. This study was partly funded by EU Horizon 2020 DOMINOES Project (grant number: 771066).

References

- [1] G. Dileep, "A survey on smart grid technologies and applications," *Renewable Energy*, vol. 146, no. 1016, pp. 2589–2625, 2020.
- [2] Y. Chen, J. Sun, Y. Yang, and J. Liu, "PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, 2021.
- [3] F. Li, Zh. Liu, T. Li, H. Ju, and W. Hua, "Privacy-aware PKI model with strong forward security," *International Journal of Intelligent Systems*, 2020.
- [4] Y. Chen, S. Dong, T. Li, Y. Wang, and H. Zhou, "Dynamic multi-key FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, vol. 16, 2021.
- [5] A. Zhao, J. Li, and M. Ahmed, "Spidernet: a spiderweb graph neural network for multi-view gait recognition," *Knowledge-Based Systems*, vol. 206, Article ID 106273, 2020.
- [6] A. Zhao, J. Dong, J. Li, L. Qi, and H. Zhou, "Associated spatio-temporal capsule network for gait recognition," *IEEE Transactions on Multimedia*, vol. 24, 2021.
- [7] A. Zhao, J. Li, J. Dong, L. Qi, Q. Zhang, and N. Li, "Multimodal gait recognition for neurodegenerative diseases," *IEEE Transactions on Cybernetics*, 2021.
- [8] X. Du, S. Tang, Z. Lu, J. Wet, and K. Gai, "A novel data placement strategy for data-sharing scientific workflows in heterogeneous edge-cloud computing environments," in *Proceedings of the IEEE International Conference on Web Services (ICWS)*, pp. 498–507, Beijing, China, October 2020.
- [9] X. Du, J. Xu, and W. Cai, "OPRC: an online personalized reputation calculation model in service-oriented computing environments," *IEEE Access*, vol. 7, no. 1109, Article ID 87760, 2019.
- [10] S. Zeadally, A. S. K. Pathan, C. Alcaraz, and M. Badra, "Towards privacy protection in smart grid," *Wireless Personal Communications*, vol. 73, no. 1007, pp. 23–50, 2013.
- [11] M. Z. Gunduz and R. Das, "Cyber-security on smart grid: threats and potential solutions," *Computer Networks*, vol. 169, Article ID 107094, 2020.
- [12] M. Gough, S. Santos, T. Alskaf, M. S. Javadi, and R. Castro, "Preserving privacy of smart meter data in a smart grid environment," *IEEE Transactions on Industrial Informatics*, vol. 18, 2021.
- [13] Z. Wang, Y. Liu, Z. Ma, X. Liu, and J. Ma, "Lipsg: lightweight privacy-preserving q-learning-based energy management for the iot-enabled smart grid," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3935–3947, 2020.
- [14] G. Tsaousoglou, K. Steriotis, N. Efthymiopoulos, and P. Makris, "Truthful, practical and privacy-aware demand response in the smart grid via a distributed and optimal mechanism," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3119–3130, 2020.
- [15] S. Uludag, S. Zeadally, and M. Badra, "Techniques, taxonomy, and challenges of privacy protection in the smart grid," *Privacy in a Digital, Networked World*, Springer, Berlin, Germany, pp. 343–390, 2015.
- [16] S. Zhao, F. Li, H. Li, R. Lu, and S. Ren, "Smart and practical privacy-preserving data aggregation for fog-based smart grids," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 521–536, 2020.

- [17] W. Zhang, Z. Guo, and N. Li, "A blind signature-aided privacy-preserving power request scheme for smart grid," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 9988170, 10 pages, 2021.
- [18] S. Li, K. Xue, Q. Yang, and P. Hong, "PPMA: privacy-preserving multisubset data aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 462–471, 2017.
- [19] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *Proceedings of the First IEEE International Conference on Smart Grid Communications*, pp. 327–332, Gaithersburg, MD, USA, October 2010.
- [20] Y. Yuan, Q. Li, and X. Han, "Efficient identity-based group blind signature scheme," *Computer Applications and Software*, vol. 27, no. 8, pp. 41–43, 2010.
- [21] Ch. Zhao, H. Yu, and J. Li, "Universally composable group blind signature," *Application Research of Computers*, vol. 34, no. 10, pp. 3109–3111, 2017.
- [22] W. Kong, J. Shen, P. Vijayakumar, Y. Cho, and V. Chang, "A practical group blind signature scheme for privacy protection in smart grid," *Journal of Parallel and Distributed Computing*, vol. 136, pp. 29–39, 2020.
- [23] M. Kumar, S. Chand, and C. P. Katti, "A secure end-to-end verifiable internet-voting system using identity-based blind signature," *IEEE Systems Journal*, vol. 14, no. 2, pp. 2032–2041, 2020.
- [24] J. Zhang, Y. Yang, and S. Xie, "A third-party e-payment protocol based on quantum group blind signature," *International Journal of Theoretical Physics*, vol. 56, no. 9, pp. 2981–2989, 2017.
- [25] Y. Jiang, S. Ge, and X. Shen, "AAAS: an anonymous authentication scheme based on group signature in VANETs," *IEEE Access*, vol. 8, Article ID 98986, 2020.
- [26] X. Zhang, J. Zhang, and S. Xie, "A secure quantum voting scheme based on quantum group blind signature," *International Journal of Theoretical Physics*, vol. 59, no. 3, pp. 719–729, 2020.
- [27] R. Xu, L. Huang, W. Yang, and L. He, "Quantum group blind signature scheme without entanglement," *Optics Communications*, vol. 284, no. 14, pp. 3654–3658, 2011.
- [28] G. Liu, W. Ma, H. Cao, and L. D. Lyu, "A novel quantum group proxy blind signature scheme based on five-qubit entangled state," *International Journal of Theoretical Physics*, vol. 58, no. 6, pp. 1999–2008, 2019.
- [29] H. Zhu, Y. Tan, X. Zhang, L. Zhu, and C. Zhang, "A round-optimal lattice-based blind signature scheme for cloud services," *Future Generation Computer Systems*, vol. 73, pp. 106–114, 2017.
- [30] P. Zhang, H. Jiang, Z. Zheng, P. Hu, and Q. Xu, "A new post-quantum blind signature from lattice assumptions," *IEEE Access*, vol. 6, Article ID 27251, 2018.
- [31] K. Tiliwalidi, J. Zhang, and S. Xie, "A multi-bank E-payment protocol based on quantum proxy blind signature," *International Journal of Theoretical Physics*, vol. 58, no. 10, pp. 3510–3520, 2019.
- [32] A. Lysyanskaya and Z. Ramzan, "Group blind digital signatures: a scalable solution to electronic cash," in *Proceedings of the International Conference on Financial Cryptography*, pp. 184–197, Anguilla, February 1998.
- [33] K. G. Paterson, "ID-based signatures from pairings on elliptic curves," *Electronics Letters*, vol. 38, no. 18, pp. 1025–1026, 2002.

Research Article

IIDQN: An Incentive Improved DQN Algorithm in EBSN Recommender System

Jianan Guo ¹, Yilei Wang ^{1,2}, Hui An,¹ Ming Liu,¹ Yiting Zhang,¹ and Chunmei Li¹

¹School of Computer Science, Qufu Normal University, Rizhao, China

²Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin, China

Correspondence should be addressed to Yilei Wang; wang_yilei2019@qfnu.edu.cn

Received 27 August 2021; Revised 4 October 2021; Accepted 25 March 2022; Published 17 May 2022

Academic Editor: Yuling Chen

Copyright © 2022 Jianan Guo 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.

Event-based Social Networks (EBSN), combining online networks with offline users, provide versatile event recommendations for offline users through complex social networks. However, there are some issues that need to be solved in EBSN: (1) The online static data could not satisfy the online dynamic recommendation demand; (2) the implicit behavior information tends to be ignored, reducing the accuracy of the recommendation algorithm; and (3) the online recommendation description is inconsistent with the offline activity. To address these issues, an Incentive Improved DQN (IIDQN) based on Deep Q-Learning Networks (DQN) is proposed. More specifically, we introduce the agents to interact with the environment through online dynamic data. Furthermore, we consider two types of implicit behavior information: the length of the user's browsing time and the user's implicit behavior factors. As for the problem of inconsistency, based on blockchain technology, a new activities event approach on EBSN is proposed, where all activities are recorded on the chain. Finally, the simulation results indicate that the IIDQN algorithm greatly outperforms in mean rewards and recommendation performance than before DQN.

1. Introduction

EBSN, event-based social networks, is a new type of social network, which connects strangers through online events recommendation. These events and activities enrich the users' experience of offline activities and broaden the social scope of users. Namely, individual interest needs could be satisfied with EBSN, so that everyone can sponsor activities offline or participate in other people's activities based on their interests, such as language learning, sports, travel, reading, etc. Therefore, EBSN expand individuals' social network. Correspondingly, with the continuous development of big data and artificial intelligence technologies, online network recommendations and evaluation feedback are becoming more influential on offline social activities. However, the current recommendation needs of EBSN could not be satisfied by traditional recommendation system technology. Liao [1] pointed out three main challenges in EBSN: existing recommendation algorithms cannot respond to the event evaluation feedback due to the lack of display

preferences in the EBSN; the data-sparse problem is still severe; and the description of activity events in EBSN is complex and diverse, with high-dimensional requirements for preferences. That is to say, considering some implicit information can increase the probability of a more accurate recommendation. Accordingly, reinforcement learning for the recommendation is considered to be applied in EBSN recommendations.

1.1. Recommendation System and Reinforcement Learning. DQN serves as an off-policy strategy combining the neural network in deep learning with the Q-learning algorithm in reinforcement learning. The Google DeepMind team first published a paper on playing Atari with deep reinforcement learning in 2013 [2, 3]. In this paper, deep learning was for the first time linked with reinforcement learning. Q-learning algorithm [4] (the Q-learning algorithm uses the maximum Q value in the Q table to select the action with the best future return, Q-table consists of all states s and all actions a in the

current state s) and neural networks are applied to calculate the Q value (The Q value is used to evaluate the value of the agent's choice of action a in state s). At the same time, the experience replay buffer is used to solve uneven data distribution. Integrating reinforcement learning and artificial neural networks allows the machine to learn from its previous experience and improve continuously. Therefore, these experiences learned through deep reinforcement learning can learn more strategies that users cannot capture under hidden behaviors. Since the recommendation of the event networks is often related to the user's activities in the recent period, static view recommendations are used in most current recommendation models, which ignore the fact that event recommendation is a dynamic sequential decision process. To overcome this drawback, reinforcement learning is applied as an attempt to recommend activities. Wu X [5] points out that compared with traditional collaborative filtering algorithms, reinforcement learning algorithms can not only easily handle the problem of large discrete state action data but also take into account the impact of users' real-time data changes. Therefore, recommendation models based on reinforcement learning are increasingly being studied by people in recent years.

1.2. Related Work. The current recommendation methods using deep learning in EBSN can discover the potential feature information in the recommendation, turning specific features into abstract features. Recently, the research of reinforcement learning in recommendation algorithms mainly includes two aspects: One is based on the input data of the recommendation system, which is divided into methods using user content information [6, 7] and methods not using user content information [8]; the other is based on the output data of the recommendation system, predicting the method of item ranking, respectively [9, 10], and the method of predicting the user scoring of items [11, 12]. Wang and Tang [13] constructed an Event2Vec model using spatial-temporal information to optimize the recommendation in EBSN. Wang et al. [14] used CNN with word embedding to capture the contextual information in EBSN but only used word embedding without considering the recommendation impact of other factors. Luceri et al. [15] used the DNN framework to predict social behavior in EBSN. We incorporate these algorithms into the consideration of the recommended results and tested them in experiments.

However, although the above solutions have been significantly improved in terms of recommendation in EBSN, there are still better solutions to optimize these algorithms, such as reinforcement learning. There are many applications of the DQN algorithm of reinforcement learning in the recommendation. Chen [16] uses a value-based DQN algorithm to recommend tips, but he only uses the keywords in the search as the feature value. He does not take into account the impact of other features like hidden features on the recommendation. Zheng [17] uses DQN to construct a DR-based IRS for news recommendations. Similarly, in another DQN-based IRS proposed by Zhao et al. [18], two

separate RNNs capture sequential positive and negative feedback. However, value-based models are not easy to handle when the action state space is vast [19].

With the continuous development of blockchain technology, various technologies in the blockchain provide a comprehensive guarantee for the security of large complex heterogeneous networks. We consider a variety of data security and integrity technologies: such as encryption mechanisms to ensure data integrity [20–23]. Defend and analyze security from the perspective of game theory [24]. Therefore, the use of blockchain technology in the EBSN network is a good way to ensure the overall security in EBSN.

1.3. Motivations and Contributions. There are some issues that need to be addressed in the recommendation system, such as a lack of dynamic recommendation, the ignorance of user's implicit behavior information as well, as the urgent need to improve online data security (e.g. inconsistency). In this paper, we focus on the above three issues. Firstly, the DQN algorithm in the reinforcement learning is introduced into the recommendation, and applied in the online activity recommendation by using the good interaction between the agent and the environment. In the proposed algorithm, the agent is regarded as the recommendation system, and the interaction process between the user and the recommendation system is regarded as the interaction process between the agent and the environment, which reflects the dynamic characteristics of the recommendation process. This avoids the drawbacks of traditional recommendations that only rely on user historical data recommendations. Secondly, in order to reflect the effects of implicit information on the recommendation algorithm, we introduced the concept of time parameters. Assign a value to the interest of a certain activity information based on the user's browsing time to identify the real intention of the user's browsing activity, thereby removing irrelevant data from the sample data. Finally, Blockchain is introduced to provide a mechanism due to the urgent requirement of data consistency. This mechanism guarantees the accuracy of online recommendations by constraining the event sponsor's event descriptions in an honest and reliable way, thus promoting the organic combination of online recommendations and offline activities.

The main contributions are as follows:

- (1) The idea of a deep Q-networks algorithm in reinforcement learning is applied to the recommendation problem of EBSN to avoid the problems of sparse matrix and poor interaction in traditional networks. Furthermore, compared with existing methods, experiments' results show that the recommendation algorithm using reinforcement learning can get higher rewards than other recommendation algorithms.
- (2) Considering the user's implicit interest, an IIDQN algorithm is proposed to improve the DQN algorithm from two perspectives: Identifying hidden nodes in the neural networks which represent implicit interest, and incentivizing those hidden nodes;

a parameter related to browsing time is added in the reward calculation, and the users' interest in the activity is explicitly demonstrated by the browsing time. Experiments' results show that the mean reward and accuracy obtained by IIDQN are significantly better than those of the DQN algorithm.

- (3) A framework is proposed based on blockchain to ensure honest behaviors for each user in the EBSN networks. This framework restricts all members in the event networks to publish "honest" offline activities in accordance with the activity information.

2. An Example of the Inactive Improved DQN

For the event recommendation part of event participants, we discuss the following issues and clarify that using reinforcement learning for event recommendation could easily infer some hidden behaviors of users. The following examples are precedent descriptions of the problems in the EBSN recommendation.

2.1. Finding Hidden Information Points of User Implicit Behaviors in Recommendation. We take the social event in Tokyo, Japan, in the Meetup as an example to analyze the implicit information in EBSN.

It can be seen from Figure 1 that the Meetup event includes time, location, traffic point, the event object, and event content. Generally, users can filter dislike events based on displayed tags or keywords. Namely, filter out specific information through tags. Simultaneously, there is still some hidden information in the activity. For example, the note part of the event plan lists the nationality ratio of people participating in the event—60% of Japanese locals, and 40% of people from other countries. Although this note may seem a trivial point for a dinner and friendship event, it is possibly of great value in an event that has other language learning needs. This means that not only classroom-type language learning can be recommended, but nonclassroom-type learning scenarios can still be implemented. However, the learning way in this environment will not be found in traditional semantic or keyword-based recommendations. The limitations in the traditional event recommendation invisibly limit people's social choices.

In conclusion, the purpose of the event is to meet different user needs. Given the traditional recommendation algorithm like a content recommendation or collaborative filtering recommendation, it is hard to balance the influence of distinct individual user preferences on implicit information. Therefore, this paper uses IIDQN, a method in reinforcement learning algorithm, to find hidden nodes in user behavior through neural networks. In this way, the implicit information in the user's interest is obtained. As in this example, a Japanese with foreign language learning needs, besides caring about his or her language learning events, will gradually pay attention to language-related activities in his or her browsing trajectory, for example, activities involving foreigners. This attention does not belong

to any interest point in the recommendation history. But it can be seen as an implicit activity recommendation point. Users' social choices will be expanded gradually.

2.2. Sample Noise Problem. The recommended sample data sets are often from a wide range of sources. The EBSN website is generally based on user clicks, browsing time, user feedback, user secondary participation rate, and so on. But for recommendations based on page clicks and browses, there is often a lot of data noise in the data set. For example, mistaken clicks caused by the user's hand sliding, pages with attractive titles or cover pages that attract users, special activities bound pop-ups carried out by website operators, etc. These data are called sample noise. Although the amount of sample noise data is not large, some websites have more sources of sample noise data. A parameter related to browsing time is added in the reward calculation, which excludes click data that has nothing to do with the user's real browsing behavior.

3. IIDQN: Incentive Improved DQN

3.1. Definition of EBSN Networks. The concept of EBSN is first proposed in Ref. [1], which is expressed as a heterogeneous network that includes online and offline relationships $G = \langle U, e_1, e_2 \rangle$, where $U \in U\{u_1, u_2, \dots, u_n\}$: represents a collection of all users.

- (1) $E_1 \in e_{\text{online}}\{e_1, e_2, \dots, e_n\}$, which means the collection of all online users
- (2) $E_2 \in e_{\text{offline}}\{e_1, e_2, \dots, e_n\}$, which means the collection of all offline users

It can be simply regarded as online and offline parts of the networks: $G_1 = \langle U, e_1 \rangle$ $G_2 = \langle U, e_2 \rangle$

Liao et al. divided the framework of the EBSN recommendation system into three layers [1]: data collection layer, data processing layer, and recommendation generation layer. The data collection layer is used to obtain various data information. The data processing layer performs pre-processing operations on the data. The recommendation layer recommends the system according to different recommendation algorithms. Compared with traditional social networks, EBSN has the following characteristics: events and user interests have a heavy-tailed distribution, event participation is heavily dependent on location characteristics, event life cycles are short, missing user display preferences, online networks are more densely distributed than offline networks, and so on. Based on these characteristics, we use IIDQN in reinforcement learning to solve recommended update timeliness and short event declaration period in EBSN.

3.2. Algorithm Calculation Equation. We associate the EBSN recommendation model with the reinforcement learning model. Reinforcement learning defines agent and environment. The agent perceives the environment and rewards

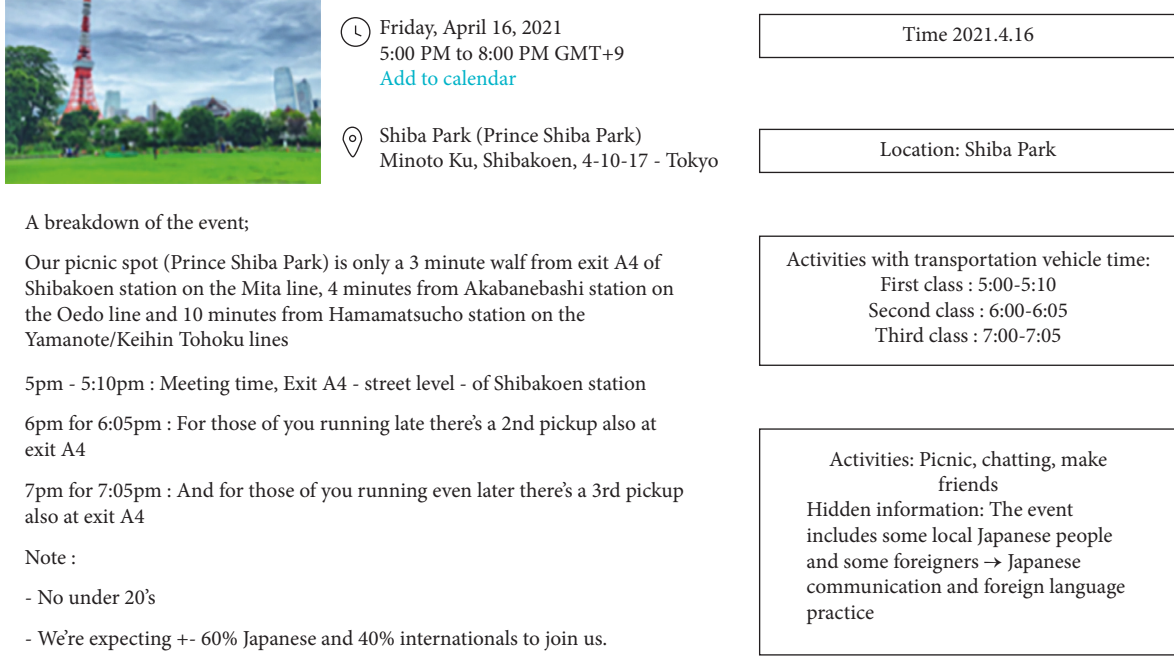


FIGURE 1: Recommended accuracy problem description diagram.

given by strategy changes, learns, and makes the next decision through environmental changes. The ultimate goal of reinforcement learning is to obtain the optimal strategy.

Currently, we define the triples in reinforcement learning. Formally, reinforcement learning consists of a tuple of three elements (S, A, R) as follows:

S is a state space, $S_t \in S$ represents the state in time t , which comes from the user's previous historical information records.

A is an action set, $a_t \in A$ represents the user's recommended choice at time t .

R is a reward matrix, $r(S_{t+1}|S_t, a_t)$ is a direct reward to Agent in state transition probability $P(S_{t+1}|S_t, a_t)$.

In IIDQN, we divide the reward value into two parts: user browsing click rewards and the length of browsing time rewards.

In this system model, we regard the recommendation part in EBSN as an Agent. The interaction process between the recommender system and the person is regarded as the change process of the Agent and the Environment. This interaction process is embodied as a Markov decision process. When the state sequence $S = \{s_0, s_1, s_2, \dots, s_t\}$ satisfies $P(S_{t+1}|S_t) = P(S_{t+1}|S_t, S_{t-1}, \dots, S_0)$, the state at the next moment merely depends on the state at the current moment.

In this model, we use the improved IIDQN algorithm based on DQN that combined Q-learning algorithm and neural network as the algorithm for the agent to accept environmental changes. Q-learning algorithm is temporal-difference learning in reinforcement learning. Temporal-difference learning can solve the model-free sequence decision problem in the Markov decision process. In the recommended situation, it is often difficult to know the state

transition probability of all actions in each state. Therefore, it is a wise choice to use the Q-learning algorithm. Its overall goal is to obtain a reward by simulating a sequence, and to obtain the maximum expected return $V(s)$ in this state by maximizing the reward:

$$V(s) = E\left(\sum_{t=0}^{T-1} \gamma^t r_{t+1} | s_0 = s\right). \quad (1)$$

Equation (1) is called the value equation, where $\gamma \in [0, 1]$ represents the discount rate. When γ approaches 0, it means that the agent is concerned about short-term returns, and when γ approaches 1, it means that the agent is more concerned about long-term returns. Equation (1) reflects that the expected return of the current state can be expressed by the expected return of the next state. Therefore, the maximum reward obtained in the current state is calculated by calculating the reward at the next moment. In addition, in the EBSN recommendation interaction process, due to the known feedback action of the user's recommendation, namely, the action selected in this state in each round of state transition is known. Therefore, we introduce the Q function of the strategy π to consider the action in the current state. The difference between the Q function and the value equation is that the Q function determines the action a in a specific state:

$$Q^\pi(s_t, a_t) = E[r_1 + \gamma Q^\pi(s_{t+1}, a_{t+1}) | s_t, a_t]. \quad (2)$$

The equation (2) is Q value function. Equation (2) determines the action in a state s_t which is related to the future state.

3.3. Markov Modelling. In order to use the reinforcement learning algorithm to address the event recommendation problem in EBSN, the recommendation problem is modeled first. The Markov recommendation conversion between simple events in EBSN is shown in Figure 2. Table 1 explains the corresponding state, action, and reward value.

In Figure 2, S_0 represents the initial state. The recommendation system acts as an agent and the user acts as the environment. In this state, the recommendation system recommends events activities to users. If the user is interested in a certain event and clicks to view the behavior, we will give a certain reward value. On the contrary, if the user ignores the recommendation and browses through search or other categories, it indicates that the user is not interested in the current recommended event. A slight penalty will be given at this time. Therefore, when making a policy selection to the system later, it is very likely that the system will no longer recommend this type of event activity to the user. When a user sees an event that fits the user's interest during the browsing process, and is ready to join the event, the recommendation system to the user is in line with the user's interest. In this state, the reward for the recommendation is great, such as the reward is 10. Therefore, the recommendation system will recommend events with similar characteristics to the event in the next recommendation.

3.4. Redefinition Reward Calculations Based on Time Parameters. In order to distinguish whether the user's browsing behavior comes from their real preferences (that is, the question raised in Problem Description 2), this paper considers the influence of browsing time on the accuracy of recommendation. In addition, the reward function is defined as a linear function related to browsing time. According to different browsing times, the correct recommendation situation is redefined. The reward value will continue to accumulate over time. In most cases, users browse according to their desire to choose their interests. However, it cannot be ruled out that users are affected by the sample space due to title interests, image interests, or other wrong click operations. Therefore, giving a certain reward value to the browsing time can distinguish the error caused by the user's mistaken click operation during the click and browse process. To a certain extent, it solves the reward calculation problem caused by the wrong click operation. Figure 3 shows a schematic diagram of the overall algorithm flow with IIDQN's mutual correspondence between agents, environments, and states.

The reward in the browsing state is linearly added to the user's scrolling browsing time, and r_t represents the additional reward based on the browsing time when the user performs the browsing state. In other words, the total reward for browsing a single event is:

$$r = \alpha * t (r < 6). \quad (3)$$

Among them, α is the reward coefficient, which means that the reward value obtained with the increase of scrolling time increases gradually, and the total r of the additional

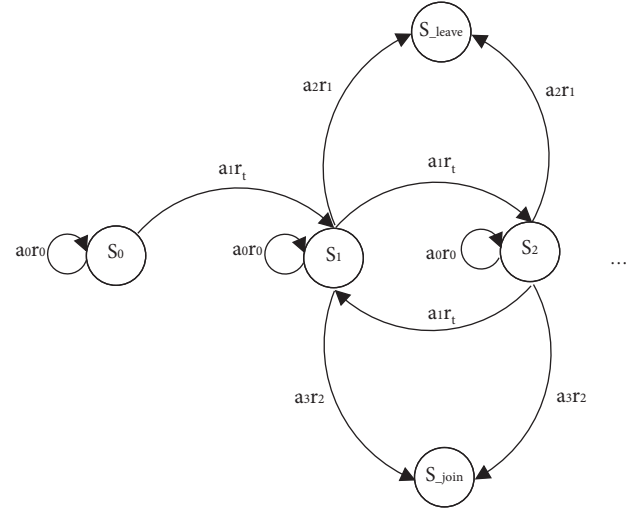


FIGURE 2: Markov state diagram.

TABLE 1: Interpretation table of the Markov state diagram.

Status set	
S_0	Initial status
S_1	Browse status
S_2	Browse status
...	...
S_{leave}	Leave website status
S_{join}	Add activity status
Action set	
a_0	No browsing
a_1	Browse recommendations
a_2	Leave the website
a_3	Join the activity
Reward set	
r_0	0
r_1	F02D 10
r_2	10
r_t	$\lambda * t$

reward value and the original reward cannot exceed a certain window value. Assuming the window value is set to 6 in the initial state, the purpose of setting the window value is to make the upper limit of the browsing time reward not exceed the reward obtained by joining the event.

Table 2 represents a simple reward calculation process:

Assuming four recommended events under this recommendation model, four simple events $e_1, e_2, e_3,$ and e_4 are recommended in the initialization state. Environment perception and selection action mean that the user chooses to browse e_2, e_4 according to his or her interests and other attributes finally join event e_2 . In this event, the state S represents the recommendation of the four events, and the environmental action A is that the user is in the state S_0 . The user makes the actions of browsing e_2, e_4 as the environment and transitions to the state S_1 . The reward obtained is the reward $R = \alpha * t_{e_2} + \alpha * t_{e_4}$ obtained according to the browsing process of e_2 and e_4 .

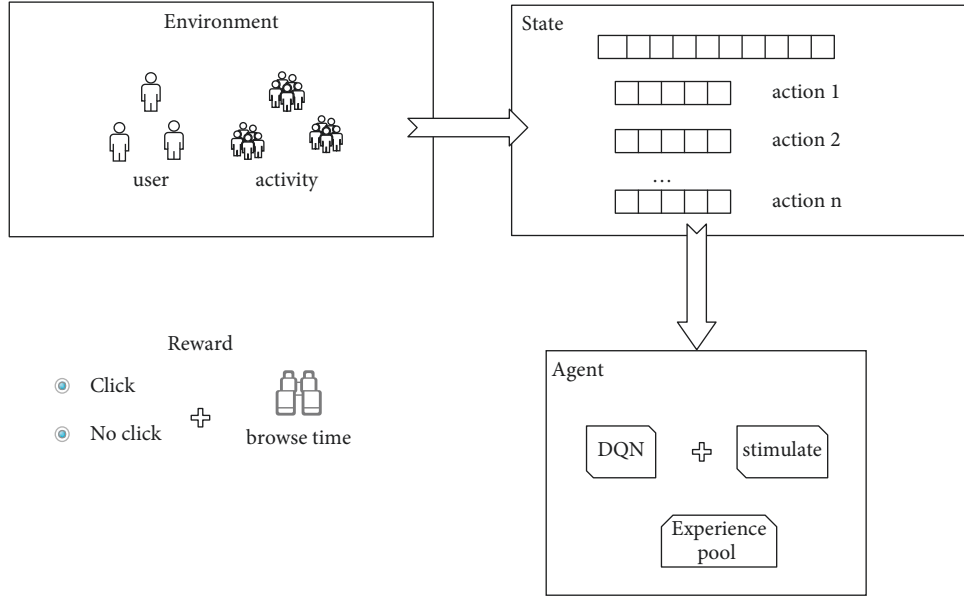


FIGURE 3: Schematic diagram of algorithm model.

TABLE 2: Event example.

	$a = \text{browsing}$	$a = \text{join event}$
e_1	-1	0
e_2	$r = \alpha * t$	+10
e_3	-1	0
e_4	$\alpha * t$	0

3.5. Reward Calculation in the Case of Feature Sparse. In order to solve the problem of the sparse number of sample features in the sample, this paper redefines the reward function in Q-network, hoping to mine the hidden information in the event to recommend the user (the impact on problem description 1). This article analyzes the initial situation when the hidden features appear, offering additional rewards under different conditions where the hidden features just appear. Therefore, weight value is constantly changing during the weight update process of the neural networks, which represents the influence of a certain feature on the final recommendation result. Specifically, we store the weight of each iteration in a matrix to calculate the weight change rate of the K iteration processes. If the rate of change rises rapidly, indicating that it is a hidden feature, a slight reward is given according to equation (4). Contrarily, if the rate of change decreases slowly, it indicates that it is not a hidden feature but may be an error value. We give a slight penalty for this change. The newly appearing networks node with a smaller weight is used as a hidden feature for additional rewards. Suppose the minimum weight characteristic value is β , and the normal sample characteristic value B has:

$$R(s_t, a_t) = \begin{cases} \frac{\alpha}{b}, & s \in D, \\ -\frac{\alpha}{b}, & s \in D', \end{cases} \quad (4)$$

where D represents the correctly classified sample set, and D' represents the incorrectly classified sample set. When the b of a certain type of sample is close to the minimum sample characteristic value β and much smaller than the normal characteristic value B :

$$B \gg b \geq \beta + \varepsilon (\varepsilon \text{ is a very small number}). \quad (5)$$

We use equation (3) to calculate the reward. Let $1/b \in (0, 1]$, where the sparse reward value increases with the sparseness of the feature value. When the sparse critical value is reached, the reward obtained is infinitely close to α .

The abscissa represents the sparseness of the interest feature, and the ordinate represents the reward value obtained in the range of the feature.

3.6. Overall Calculation Flow of IIDQN Algorithm. This section describes the algorithm process of the deep Q network using reinforcement learning based on incentive improvement in the recommendation system.

Figure 4 shows the specific process implementation scheme of IIDQN. Next, we discuss the specific algorithm implementation. Figure 4 shows the overall calculation process of the IIDQN algorithm.

In this model, in the initialization state, we define an experience pool and two networks with the same structure. [lines 1,2] One is called the Q network for each round of model iteration calculations, and the other network is called the target network. The parameter value in the target network is used as the final calculated true value. Select the recommended item a [lines 3,4] in the initialization state s . Next, proceed to the part of the recommendation system interacting with the user [line 5]. The reward function of this action will take into account the time incentives mentioned in Section 3.4. Then, put the quadruple (S, A, R, S') into the experience pool. Sample the experience in the experience

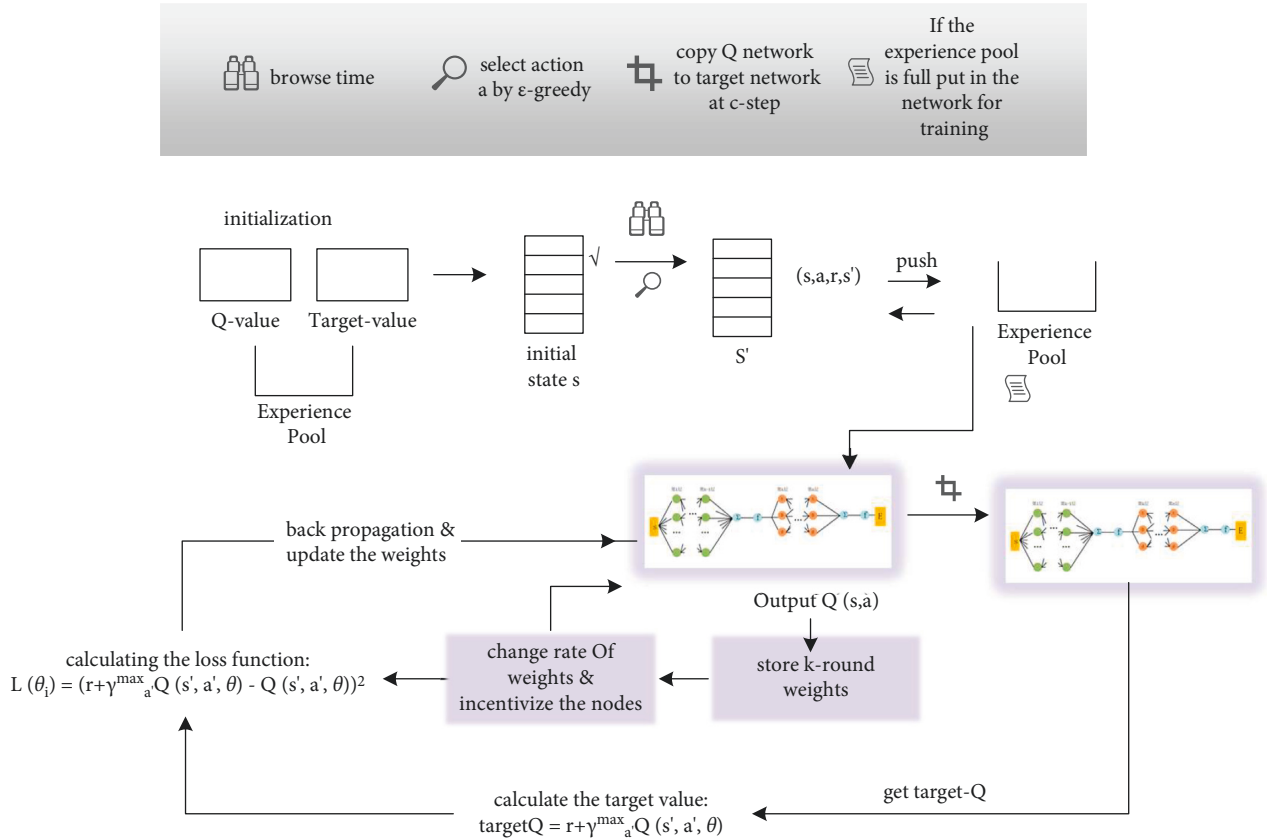


FIGURE 4: The calculation process of IIDQN in EBSN.

pool through experience replay [lines 6–7]. During Q network training, N pieces of experience are randomly selected from the experience pool and placed in the network for experience selection. That is, the network input of the Q network is in N states s. The final output value obtained by the Q network is the expected reward Q value of all actions that can be selected in each state. Accordingly, the Q-value is calculated through the neural network, and the Q network is recalculated using the square loss function [line 8]. Identify sparse nodes and incentivize the sparse weights in the range [line 9]. Redefine a few classified samples in the data set by storing feature values with smaller weights in nodes. The change rate is dynamically calculated for each round of the weight characteristic value, and the nodes with smaller weight values are stimulated. In this way, implicit information is recognized. In order to maintain the stability of the calculation results in the network, the parameters in the Q network are copied to the target network for storage processing every c step [line 10]. The calculation of the target value is performed according to the stored Q value in the target network, and the calculation process is the calculation process of the true value in the Q-learning algorithm. The specific algorithm flow is shown in Algorithm 1, where the bold part is the difference between the IIDQN and DQN algorithms:

4. Experiment Results and Analysis

4.1. Experiment Data Description. The recommendation problem of a single user in the event network is analyzed. The data of this experiment are from the data set of the meetup website. The recommendation changes with the user’s preferences, and finally mean reward is obtained.

This data set classifies the 36 types of activity group interests in a meetup in detail and divides the browsing time of each user’s activity browsing event in detail.

Behavioral strategy: the experiment’s DQN algorithm strategy based on reward transformation adopts the epsilon-greedy exploration strategy.

Next, we discuss several important related parameters. For example, in Section 3 we introduce parameter α . The reward factor can choose to be 0.05, that is, if the user’s browsing time is 120 s, the reward value is 6. We use the user’s browsing time of 120 s as the limit, and set the reward threshold to 6. Over 120 s, we default that the user is interested in the current browsing, and no additional bonus value will be added. In addition, we use the ϵ -greedy strategy to explore the information in the experience pool, by doing so to ensure the recommendation results are independent and identically distributed. The initial exploration ϵ is 0.6, and the coefficient will continue to decrease as the agent

IIDQN algorithm

Inputs: state space S , action space A , discount rate γ , learning rate α , parameter update interval C .

- (1) Randomly initialize the parameters θ of the Q-networks and randomly initialize the parameters θ of the target Q-networks.
- (2) Initialization state S .
- (3) Select action a .
- (4) **Perform action, get reward r and next action S_0 through environment.**
- (5) Put S, a, r, S_0 into the experience pool and sample
- (6) Sample ss, aa, rr, ss_0 in the experience pool for the next action. Let y be the target value

$$y = \begin{cases} rr, & ss' \text{ is the termination state} \\ rr + \gamma \max_{a'} Q(ss', a'), & \text{or} \end{cases}$$

- (7) Update the weights by back propagation mechanism and retrain the Q networks using gradient descent algorithm
- (8) **Detect sparse nodes and apply additional reward/penalty updates to nodes**

$$\mathbf{R}(s_t, a_t) = \begin{cases} a/b, & s \in D \\ -a/b, & s \in D' \end{cases}$$

- (9) Update the target networks every N steps and copy the current networks parameters to the target networks.

ALGORITHM 1: Incentive improvement algorithm based on Q networks in EBSN.

continues to learn, and the termination exploration ϵ is 0.05. This shows that more attention is paid to the exploration of newly added data in the initial stage, so the size of the DQN experience pool cannot be too small.

4.2. Comparison of Recommended Models. To find the most suitable recommendation algorithm under the EBSN model, we compared several proposed frameworks, including the original DQN algorithm. To evaluate the recommendation performance, we divide each data set into a training set and a test set, use 80% of them as the model training set, and train 20% of the model data.

CF: The collaborative filtering recommendation method mainly benefits users by displaying user information on different preferences and predicts information by finding users with similar preferences.

DNN: Deep Neural Networks, which use neural networks to predict user preferences, are also the user's historical data recommendation information, and the output is the DNN output recommendation item.

RNN: It is a type of sequence data input, recursive in the evolution direction of the sequence, and all nodes (cyclic units) are connected in a chain.

DQN: We first use the DQN algorithm for recommendation prediction, determine the correspondence between the agent and the environment, and input the user's historical information as the state.

Improved DQN: The improved DQN algorithm for incentives is proposed in Section 3 of this article.

The above types of recommendation models have a wide range of choices. There are traditional recommendation models and deep learning framework recommendations, and model recommendations in reinforcement learning. The DQN algorithm is chosen as the baseline because the DQN algorithm can continuously update the strategy during the interaction process. DNN and RNN are cited as a comparison based on the contrast gap between neural networks in the DQN algorithm. RNN can capture the time series of

the user's browsing history because the order of recommended browsing on the product page can affect each other. Therefore, we introduce RNN to consider this reason.

According to the effect of the above several recommended models on the simulator, we use NDGC [25] and MAP [26] as the two evaluation criteria for comparison. NDCG is a normalized DCG, which is an evaluation index for measuring search recommendations. This indicator takes into account the relevance of all elements. MAP (mean Average Precision) is an indicator of recommended accuracy. It is calculated by summing the mean accuracy of all categories and dividing by all categories. Figure 5.

According to the recommended data in the above figure, it can be seen that:

- (1) Generally, the recommendation efficiency of using deep learning and reinforcement learning frameworks is significantly better than the general recommendation results. To a certain extent, the traditional recommendation model represented by CF ignores the time interaction factor in the user input information. Since traditional models pay more attention to user characteristics, they are not suitable for interaction-based recommendations in EBSN.
- (2) In addition, comparing the deep recommendation model (DNN, RNN) and the reinforcement learning recommendation model (DQN), we can find that reinforcement learning still performs better than the deep learning recommendation to a certain extent. Because deep learning pays more attention to recommending activities that can increase the model's timely rewards, the model promotion in reinforcement learning will merge the user's rewards throughout the participation cycle. The DQN model focuses on the overall user experience from the beginning of user registration to a long time in the future. The internal user's overall experience will be quantified as the total revenue of the model.

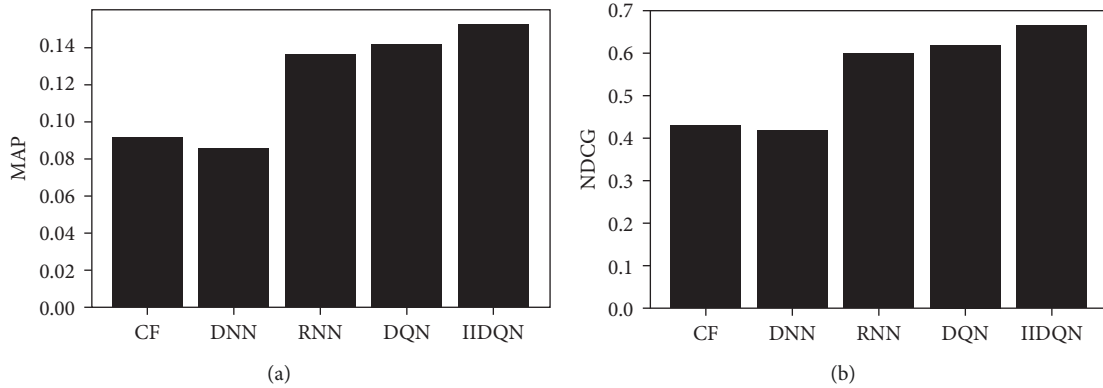


FIGURE 5: Performance comparison in different recommendation systems. (a) Overall performance comparison on MAP. (b) Overall performance comparison on NDCG.

- (3) Based on the comparison of the above recommendations, we can see that the model selected for reinforcement learning is better in our recommendation to solve this problem. Nevertheless, only using DQN cannot solve some of the problems mentioned in the previous problem description. Therefore, we finally improved the DQN algorithm and finally got a recommendation effect better than DQN.

5. Blockchain-Based Activity Methods

As the core of the blockchain, the consensus algorithm ensures the mutual trust relationship between nodes in the blockchain and thus maintains the security of the blockchain. However, offline users in the event network are mostly strangers, and it is difficult to establish a trust relationship between them. Therefore, using the characteristics of blockchain technology can guarantee the mutual trust relationship between nodes, and we consider the new problem of recommendation in EBSN: The recommended description of online activities is inconsistent with the actual offline activities. In order to solve this problem, in this section, we propose a deposit consensus mechanism based on blockchain technology. The problem of whether the activities in the EBSN network conform to the activity recommendation is modeled on the blockchain system, and the deposit consensus mechanism is used to solve the problem.

5.1. Model Overview. Based on the scattered and complex characteristics of event network nodes, blockchain technology is applied to the event network, and the entire network is regarded as a scattered blockchain node. The behavior information generated by all users in the event network will be written on the chain for recording.

There are two kinds of membership in the network: sponsor and participant. Correspondingly expressed as two kinds of user nodes on the blockchain, sponsor is the event initiator of each activity. Sponsor needs to obtain the consent of a few validators before proceeding with the actual event initiation. Validator is a few randomly generated validators in

the chain that are used to verify the identity of the sponsor and vote whether the activity proposed by the sponsor is on the chain. These few randomly generated nodes are equivalent to the identities of temporary supervisors, ensuring the fairness and security of activities among nodes in the entire network. When the sponsor creates an activity, a new consortium chain is generated. The address of the consortium chain and the users' name that caused him or her to be generated will be recorded on the public chain. Each block on the alliance chain records an event activity information, which includes the trust deposit of the organizer, the overall process recorded in the activity event, and the activity transaction fee submitted by the user. Figure 6 shows the main activity function of event activity group A in the EBSN blockchain network.

For other user participants in the chain, when participants want to participate in an activity, they will apply to the legal activity sponsor to join the group in the same way. After the validator in the activity group agrees to join the group, all activity transactions and activities during the activity will be written on the alliance chain within the organization.

5.2. Build Model. We construct the current activity relationship in the event network as a network relationship in the blockchain. In addition, the set $E = \langle b, U, A \rangle$ in the event network. Among them: b represents the block number, $U \in \{U_s, U_p\}$ represents the set of all users and divides the set of all users in the network into two categories, event sponsor (u_s) and event participant (u_p). $A \in \{a_1, a_2, \dots, a_n\}$ represents the set of all alliance chains.

In the network, for the number of z users, user u corresponds to z nodes in the blockchain network. And, there is a public chain and multiple alliance chains in the network. The public chain records the information of all event groups, and the alliance chain records the process information of each event group in the entire event activity. It mainly contains information such as time of initiation, specific details of event activities, transaction records of activity fees for members to participate in the event, and credit deposit submitted by the event initiator, etc.

For sponsor users u_s who want to create an activity in the blockchain, they first need to publish an event activity

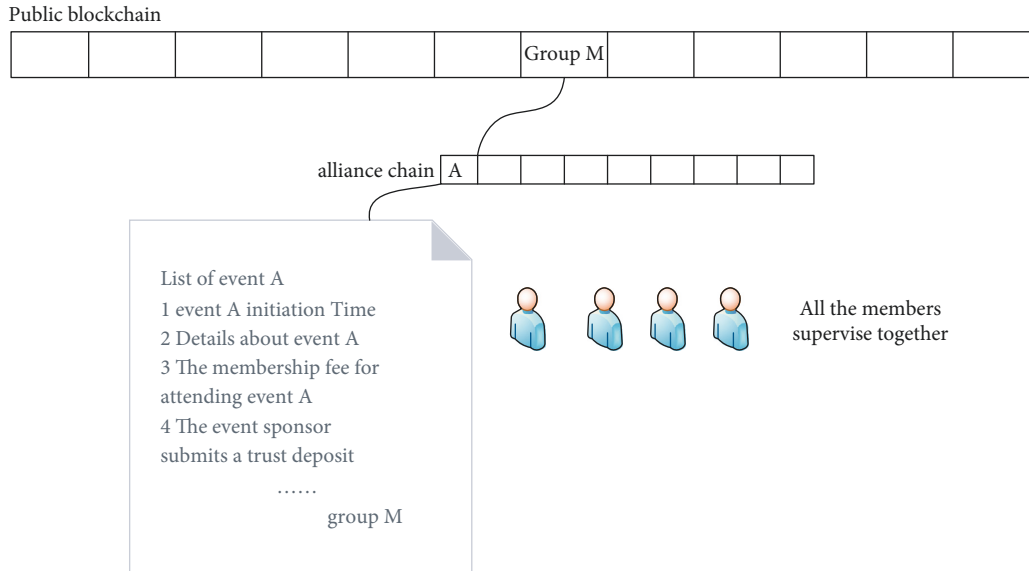


FIGURE 6: Examples of blockchain-based activities.

application information to the public chain. In addition to the user's own personal identity verification information and event activity details, the application information also contains how much money the user decides to spend as the trust deposit for this activity. If more than half of the verification nodes pass the verification and agree to the user's activity creation, then $u_s a$ will successfully create the activity event M , and will generate a consortium chain corresponding to the activity event M . On the public chain, the user information of the initiator of the event, the corresponding consortium chain address, and trust deposit will be recorded. In the alliance, it is mainly used to record the actual information experienced by the event activity offline. In this way, it not only guarantees the safe conduct of offline activities but also ensures the consistency of online event network recommendation descriptions and offline event activities.

5.3. Trust Deposit. Leverage the immutability feature of the blockchain to ensure that the online recommendations seen by participants are consistent with offline activities. For other user participants present in the chain, the event organizer will take a part of the amount as a trust deposit when creating the event, and put the trust deposit on the blockchain. After the event, each event participant will rate and score the entire event process. At the same time, the most important thing is to score whether the activity meets the description of the activity initiator on the web page. This will be used as a very important indicator for the recommendation system to make recommendations, to promote the evaluation of the overall activity experience by online recommendations. Finally, the average value of this indicator is used to quantify the event organizer. Here, we take into account the influence of different people on the evaluation preferences, and use the overall variance to process the calculation. According to the user score, it is finally determined how much the trust deposit originally placed on the chain by the initiator of the event can be recovered. If the

score is not satisfactory, the event organization is very likely to be seriously inconsistent with the original description, and then the event organizer will receive negative feedback from the participants. In addition, the recommendability of the event organizer will be weakened, and the original trust deposit on the chain will also be deducted.

In order to prevent malicious evaluation by event participants, the overall evaluation process follows the consensus protocol PBFT in the blockchain. Practical Byzantine fault tolerance (PBFT) is one of the consensus algorithms proposed earlier [27]. As a practical consensus algorithm based on state machine, PBFT's role model can correspond to the organizers and participants in the event network. Although the consensus mechanism can ignore the malicious influence of a certain user to a certain extent, it is also difficult for the algorithm to achieve consensus if malicious commenters exceed one-third of the total participants.

6. Conclusions

EBSN is a field of promising research today, which is of great significance from online and offline security research. This paper offers a mechanism based on blockchain technology in EBSN, which includes creating activities on the organizer chain and recommending online and offline event activities guaranteed by blockchain. Simultaneously, offline activities conform to the online recommendation description through the blockchain. Furthermore, we add the reinforcement learning algorithm to the event recommendation, improve the DQN algorithm, and propose IIDQN. Through this algorithm, the recommendation process of dynamic interaction can be simulated. Improve the time-related parameters to eliminate sample noise in the recall phase. However, the work done in this paper needs further research. For example, it is only compared with a few typical recommendation algorithms, and the differences between other algorithms are not considered. In addition, we merely considered the impact of time in this

study, and there are many other factors that will affect the final recommendation accuracy. For the overall algorithm, it has only been verified in a small scale. In the following work, it should be further extended to a more general situation for further analysis and research.

Data Availability

The CSV data used to support the findings of this study are available in <https://www.kaggle.com/stkbailey/nashville-meetup>.

Conflicts of Interest

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

Acknowledgments

This study was supported by the Foundation of National Natural Science Foundation of China (Grant Number: 62072273, 72111530206, 61962009, 61873117, 61832012, 61771231, and 61771289), Natural Science Foundation of Shandong Province (ZR2019MF062), Shandong University Science and Technology Program Project (J18A326), Guangxi Key Laboratory of Cryptography and Information Security (No: GCIS202112), The Major Basic Research Project of Natural Science Foundation of Shandong Province of China (ZR2018ZC0438); Major Scientific and Technological Special Project of Guizhou Province (20183001), Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No. 2019BD-KFJJ009), and Talent project of Guizhou Big Data Academy. Guizhou Provincial Key Laboratory of Public Big Data ([2018]01).




References

- [1] G. Q. Liao, T. M. Lan, and X. M. Huang, "Survey on recommendation systems in event-based social networks," *Ruan Jian Xue Bao/Journal of Software*, vol. 32, no. 2, pp. 424–444, 2021.
- [2] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing Atari with deep reinforcement learning," *Computer Science*, <https://arxiv.org/abs/1312.5602>, 2013.
- [3] V. Minh, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.
- [4] C. J. C. H. Watkins, "Learning from delayed rewards," *Robotics and Autonomous Systems*, vol. 15, no. 4, pp. 233–235, 1995.
- [5] X. Wu, Y. Dong, B. Shi, A. Swami, and N. V. Chawla, "Who will attend this event TOGETHER? Event attendance prediction via deep LSTM networks," in *Proceedings of the 18th SIAM International Conference on Data Mining (ICDM)*, pp. 180–188, University of Illinois, Chicago, IL, USA, 2018.
- [6] A. Van Den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," *Advances in Neural Information Processing Systems*, vol. 26, no. 2, pp. 2643–2651, 2013.
- [7] P. Hamel, S. Lemieux, Y. Bengio, and D. Eck, *Temporal Pooling and Multiscale Learning for Automatic Annotation and Ranking of Music Audio*, Ismir, Canada, 2011.
- [8] Y. Zuo, J. Zeng, M. Gong, and L. Jiao, "Tag-aware recommender systems based on deep neural networks," *Neurocomputing*, vol. 204, pp. 51–60, 2016.
- [9] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *Computer Science*, abs:1511.06939, 2015.
- [10] W. Caihua, J. Wang, J. Liu, and W. Liu, "Recurrent neural network based recommendation for time heterogeneous feedback," *Knowledge-Based Systems*, vol. 109, no. 1, pp. 90–103, 2016.
- [11] H. Larochelle and I. Murray, "The neural autoregressive distribution estimator," *Journal of Machine Learning Research*, vol. 15, pp. 29–37, 2011.
- [12] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proceedings of the International Conference on Machine Learning ACM*, pp. 791–798, Corvallis Oregon, June 2007.
- [13] Y. Wang and J. Tang, "Event2Vec: learning event representations using spatial-temporal information for recommendation," in *Proceedings of the 23rd Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)*, pp. 314–326, Macau, China, April 2019.
- [14] Z. Wang, Y. Zhang, H. Chen, Z. Li, and F. Xia, "Deep user modeling for content-based event recommendation in event-based social networks," in *Proceedings of the 2018-IEEE Conference on Computer Communications*, pp. 1304–1312, 2018.
- [15] L. Luceri, T. Braun, and S. Giordano, "Social influence (deep) learning for human behaviour prediction," in *Proceedings of the International Workshop on Complex Networks*, pp. 261–269, Honolulu, HI, USA, April 2018.
- [16] S. Y. Chen, Y. Yu, Q. Da, J. Tan, H. K. Huang, and H. H. Tang, "Stabilizing reinforcement learning in dynamic environment with application to online recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1187–1196, London, UK, July 2018.
- [17] G. Zheng, F. Zhang, Z. Zheng et al., "DRN: a deep reinforcement learning framework for news recommendation," in *Proceedings of the 2018 World Wide Web Conference*, pp. 167–176, Lyon, France, April 2018.
- [18] X. Zhao, L. Zhang, and Z. Ding, "Recommendations with negative feedback via pairwise deep reinforcement learning," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1040–1048, New York, NY, USA, 2018.
- [19] J. She, Y. Tong, C. Lei, and C. Chen, "Conflict-aware Event-Participant Arrangement," in *Proceedings of the IEEE International Conference on Data Engineering IEEE Computer Society*, pp. 735–746, 2015.
- [20] C. Ge, W. Susilo, and J. Baek, "Revocable attribute-based encryption with data integrity in clouds," *IEEE Transactions on Dependable and Secure Computing*, no. 1, p. 1, 2021.
- [21] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "A verifiable and fair attribute-based proxy Re-encryption scheme for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, 2021.
- [22] C. Zhao, S. Zhao, M. Zhao et al., "Secure multi-party computation: theory, practice and applications," *Information Sciences*, vol. 476, pp. 357–372, 2019.
- [23] Y. Lei, S. Chen, L. Fan, F. Song, and Y. Liu, "Advanced evasion attacks and mitigations on practical ML-based phishing website classifiers," 2020, <https://arxiv.org/abs/2004.06954>. 06954.

- [24] T. Li, Y. Chen, and Y. Wang, “Rational protocols and attacks in blockchain system,” *Security and Communication Networks*, vol. 2020, Article ID 8839047, 2020.
- [25] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002.
- [26] A. Turpin and F. Scholer, “User performance versus precision measures for simple search tasks,” in *Proceedings of the 29th Annual International ACM SI*, Seattle, WA, USA, August 2006.
- [27] M. Castro and B. Liskov, “Practical byzantine fault tolerance and proactive recovery,” *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 398–461, 2002.

Research Article

Cloud Storage Data Access Control Scheme Based on Blockchain and Attribute-Based Encryption

Xiaodong Yang ¹, Aijia Chen ¹, Zhisong Wang,¹ and Shudong Li ²

¹College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China

²Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China

Correspondence should be addressed to Xiaodong Yang; y200888@163.com and Shudong Li; lishudong@gzhu.edu.cn

Received 19 November 2021; Revised 23 December 2021; Accepted 30 March 2022; Published 11 May 2022

Academic Editor: Yuling Chen

Copyright © 2022 Xiaodong Yang 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.

Cloud storage is a popular model of the application in various fields, and the security of storage data and access permission have been widely considered. Attribute-based encryption (ABE) provides fine-grained user access control and ensures data confidentiality. However, current ABE access control schemes rely on trusted cloud servers and provide a low level of security. To solve these problems of traditional encryption schemes, we propose a blockchain-based and ABE cloud storage data access control scheme. In this article, blockchain and smart contract technology are the core elements to ensure data integrity and build a decentralized verification method for outsourcing results. This application can minimize the reliance on servers in the cloud environment. Based on the ciphertext-policy ABE algorithm, the proposed scheme supports a hidden access policy to avoid the risk of privacy leakage. In addition, we adopt outsourcing technology and predetected decryption algorithms to reduce the computational overhead of local and outsourced servers. Security analysis and performance evaluation show that our proposed scheme has high computational efficiency and satisfies the condition of indistinguishability under the chosen-ciphertext attacks.

1. Introduction

Cloud storage technology uses the storage space of cloud servers to provide powerful data storage capability [1]. Data owners can overcome the obstacle of restricted storage resources at user terminals by storing data in the cloud. Therefore, cloud storage has become more popular in various specific industries in recent years, such as the Internet of Things (IoT) [2, 3], the Industrial Internet of Things environment [4], and electronic health records [5, 6]. However, the data collected by cloud servers and IoT devices face many attacks [7] during data transmission and storage. Meanwhile, sensitive data are vulnerable to tampering or forgery attacks during the transmission via public channels, which exposes users' private information to the risk of being leaked. Therefore, it is critical to consider privacy protection and data confidentiality in the network. In the most typical schemes, encryption technology is adopted to achieve data confidentiality and privacy. To provide more detailed

privacy protection, some researchers introduce the most recent privacy protection technologies in their schemes. For instance, a location privacy protection scheme [8] anonymizes the source location, which contains significant information about the target being observed and tracked. Moreover, a homomorphic encryption scheme with higher performance [9] is proposed to achieve privacy protection of data stored in the central server.

Although the encryption mechanism can guarantee the confidentiality and privacy of the data, it does not ensure that the data are legally obtained. In cloud storage applications, the data stored in the cloud server cannot be fully controlled by the data owner. To prevent malicious users and cloud server providers from accessing data, a trusted access control mechanism is also essential.

The CP-ABE [10] not only provides data confidentiality but also allows fine-grained and flexible access control to improve the security of the data. However, the traditional CP-ABE scheme [10, 11] has some drawbacks in practical

applications. For example, the access control policy in the CP-ABE is constructed by attribute information-related users, which may contain private information about the user's identity. Second, attribute-based encryption algorithms frequently use a large number of bilinear pair computations, significantly increasing the encryption and decryption computational overhead. To reduce computational costs, on the one hand, an increasing number of schemes outsource decryption operations to third-party servers. However, few of these systems consider the correctness of calculation results from cloud servers. On the other hand, most access control schemes on cloud platforms are established using prime-order bilinearity to reduce the computational burden. This design's reduced computational burden comes at the expense of lower security, so it can only satisfy indistinguishability under chosen-plaintext attack (IND-CPA). Although there are already some schemes that can partially solve the above problems, we still need to consider some detailed and in-depth issues. The existing cloud storage access control scheme is designed based on the traditional cloud server, which increases the trusted dependence on the cloud server. Unfortunately, semitrusted cloud servers are curious about the processed data while executing user commands. If the cloud server fails unpredictably or is maliciously attacked and outputs incorrect results, it may cause users to obtain incorrect data.

Blockchain technology [12] is a widely emerging technology based on distributed ledgers that has the advantages of decentralization. However, at the same time, due to the openness of blockchain, data security and supervision are also faced with challenges [13, 14]. Therefore, the combination of blockchain technology and traditional access control is a promising structure. Blockchain technology can enhance the reliability of traditional schemes, and the encryption mechanism of the scheme can protect the data security of the blockchain. In this article, we are committed to establishing a reliable access control mechanism in an untrusted cloud environment. We propose a cloud storage access control scheme based on blockchain and attribute-based encryption, which realizes data verification and ensures the verifiability of the outsourced decryption results and the integrity of the cloud storage data in a decentralized way.

The main contributions of our proposed program are as follows:

- (i) The support of hidden access control policies reduces the risk of user privacy information disclosure in traditional CP-ABE.
- (ii) The use of smart contracts deployed on the consortium blockchain can achieve a decentralized verifiable outsourcing scheme while ensuring the integrity of data in the cloud.
- (iii) The dependence on fully trusted cloud servers in traditional cloud server-based schemes is removed by introducing blockchain technology.
- (iv) Our scheme is proven to meet CCA security under the random oracle model, which has stronger

security than similar schemes. Performance analysis shows that the new scheme has comparable computational overhead.

The rest of the article is organized as follows. Section 2 introduces the related work. Preliminary knowledge related to our scheme is described in Section 3. In Section 4, we present the system model, security model, scheme framework, and detailed construction of the proposed scheme. The correctness analysis is given in Section 5. In Section 6, we provide security analysis and security proof of the new scheme. In Section 7, we discuss the performance analysis and computational efficiency of our scheme. The work of this scheme is concluded, and the outlook is presented in Section 8.

2. Related Work

To overcome the problem of multiperson sharing of encrypted data, an attribute-based encryption system (ABE) [15] was proposed as a one-to-many encryption mechanism. More specifically, ciphertext-policy attribute-based encryption (CP-ABE) [10] allows the data owner to refine the user authority of the data visitor to the attribute level by setting a policy. In other words, CP-ABE can achieve effective fine-grained access control under the condition of ensuring data security.

However, the traditional CP-ABE Schemes [16, 17] usually publish the access policy in the form of plaintext. Anyone who obtains the ciphertext (including cloud servers) can infer part of the secret information included in the ciphertext, endangering the user's identity privacy. In addition, sensitive data must also be protected as private data in specific fields.

To address the above issues, Kapadia et al. [18] proposed a policy-hiding CP-ABE scheme. However, an online semitrusted server was introduced in [18] to reencrypt the ciphertext for each user, thus making the server a bottleneck in the entire system. Nishide et al. [19] developed two CP-ABE schemes to hide the policy, which express the access control policy through AND logic with wildcards. Based on the decisional assumption of subgroups, Lai et al. [20] suggested an adaptively secure policy hiding the CP-ABE technique over a bilinear group of combinatorial orders. Although the scheme in [20] improves security, the computational cost grows with the increase of the attributes. Hur [21] constructed a scheme that supports arbitrary expressions with monotonicity and blinds the access policy within the ciphertext. However, this scheme is proven to be secure using the generic group model, which is normally considered heuristically rather than provably secure. Afterwards, Helil Rahman [22] constructed a CP-ABE access control scheme based on the scheme in [21]. We introduce an additional entity (the SDS monitor) in [22] to handle the problem of sensitive dataset constraints, but the policy is disclosed for all entities. Song et al. [23] made improvements to the access tree on the basis of the scheme in [24] to realize policy hiding based on the access tree. Through the application of secret sharing in "and," "or" and "threshold,"

attribute values with permission are hidden in all attribute values of the system. However, as the expression ability of the access structure grows, the communication overhead also increases.

To reduce the overhead of a large number of bilinear pairings required for the CP-ABE decryption calculation, Green et al. [25] proposed a scheme with outsourced decryption. In their article, the outsourcing server uses a transformation key for decryption, which is generated by the data user. However, their scheme lacks a verification mechanism for the calculation results of the outsourcing server. Then, on the basis of the scheme in [25], Lai et al. [26] verified the result returned by the outsourcing server by adding a ciphertext component. However, at the same time, this method doubles the ciphertext length of the ABE-type and El Gamal-type encryption systems. In recent years, with the development of fog computing, fog nodes have been widely used in cloud environments. Li et al. [27] presented a verifiable outsourced multiauthorization access control method that delegated most encryption and decryption work to fog nodes. This scheme can lighten the user's processing load and verify the reliability of outsourced computing outputs. In fog-enhanced IoT systems, an access control scheme with hidden access structures and outsourcing computation was presented by [28], which uses fog nodes to conduct outsourcing decryption and verification procedures. Lin et al. [29] invented a new attribute-based scheme combined with symmetric encryption technology to achieve efficient verifiability. In addition, they presented a verifiable unified model for the OD-ABE. However, all of the abovementioned verifiable outsourcing schemes meet the CPA security requirements. A verifiable hidden policy CP-ABE with a decryption testing scheme (VHPDT) was proposed by Zhao et al. [30], which is CCA-secure. Meanwhile, the VHPDT scheme introduces a predetection algorithm to increase the efficiency of the decryption. However, this scheme does not consider the integrity verification of the data and needs to rely on trusted cloud servers. However, cloud servers cannot be completely trusted, and dangers such as user data leakage and tampering will persist.

Blockchain technology [12] is an emerging technology based on distributed ledgers that has the advantages of decentralization. Many systems [31–34] introduce blockchain into the traditional cloud server-based structure to better realize decentralized security schemes. Rahulamathavan et al. [32] proposed combining blockchain technology with ABE to realize data confidentiality and privacy protection. However, the large amount of computing overhead generated by ABE is not suitable for the resource-constrained IoT environment. Zhang et al. [33] introduced blockchain-based smart contract technology and designed a BaDS scheme in the IoT, which not only reduces the cost of decryption but also improves the flexibility of traditional CP-ABE for access control. A blockchain-based outsourcing verifiable CP-ABE scheme was offered by Zhang [34], which uses smart contracts to achieve verifiability of the outsourcing results. However, decrypting and obtaining plaintext by smart contracts will reduce the security of the system.

3. Preliminary Knowledge

3.1. Composite-Order Bilinear Group. Assuming that φ is a group generation algorithm, the input λ is a security parameter, and the output $(N = p_1 p_2 p_3, G, G_T, \hat{e})$ is a tuple, where N is the product of three prime numbers p_1, p_2 , and p_3 ; G and G_T are cyclic groups with order N ; $\hat{e}: G \times G \rightarrow G_T$ is a bilinear map satisfying the following conditions:

- (1) **Bilinearity:** for any $g_0, g_1 \in G$, $c, d \in \mathbb{Z}_N$, we have $e(g_0^c, g_1^d) = e(g_0, g_1)^{cd}$.
- (2) **Nondegeneracy:** if $x \in G$, then $e(x, x)$ has the order N in G_T .
- (3) **Computability:** if $\hat{e}: G \times G \rightarrow G_T$, then operations in G and G_T are effectively computable in polynomial time, and G and G_T are bilinear groups.
- (4) **Orthogonality:** G_{p_1}, G_{p_2} , and G_{p_3} are three subgroups of G , with the order of p_1, p_2 , and p_3 , respectively. The orthogonality of the subgroups can be known as follows:
 - (a) For any $h_{p_1} \in G_{p_1}$ and $h_{p_2} \in G_{p_2}$, then $e(h_{p_1}, h_{p_2}) = 1$.
 - (b) For any $h_{p_1} \in G_{p_1}$ and $h_{p_2} \in G_{p_2}$, where $a, b, c, d \in \mathbb{Z}_N$, equation $e(h_{p_1}^a h_{p_2}^b, h_{p_1}^c h_{p_2}^d) = e(h_{p_1}, h_{p_2})^{ab} = e(h_{p_1}, h_{p_2})^{cd}$ holds.

3.2. Discrete Logarithm (DL) Problem. Let G_0 be a multiplicative cyclic group of order p_1 and g_1 be the generator of G_0 . Given a tuple $(g_1, \Delta = g_1^x)$, where $\Delta \in G_0$, the DL problem has difficulty calculating $x \in \mathbb{Z}_N$.

3.3. Blockchain and Smart Contracts. The essential function of blockchain technology is a distributed ledger that cannot be tampered with and counterfeited [12]. Blockchain technology joins data blocks in chronological order to form a chain data structure and uses cryptography to assure the chain's immutability and security. Moreover, blockchain encourages network nodes to participate in and jointly maintain chain data by setting up incentive mechanisms to provide a reward. The consensus mechanism is adopted to ensure the fairness of transactions, which is based on multiparty consensus and will not be undermined by the complicity of a few malicious nodes. Therefore, blockchain can be used as a low-cost and highly reliable infrastructure. Blockchain is deployed in the forms of public blockchain, private blockchain, and consortium blockchain. The public blockchain is a mode in which any node is open to anyone. This mode allows everyone to participate in the calculation of this block, and anyone can download and obtain the full blockchain data. The private blockchain is a private chain in which only licensed nodes can be involved and view all data. Consortium blockchain means that the permissions of each node participating are completely equal. Without total mutual trust, each node can realize the trustworthy exchange of data, but each node often has an associated entity organization that may only join or leave the network after

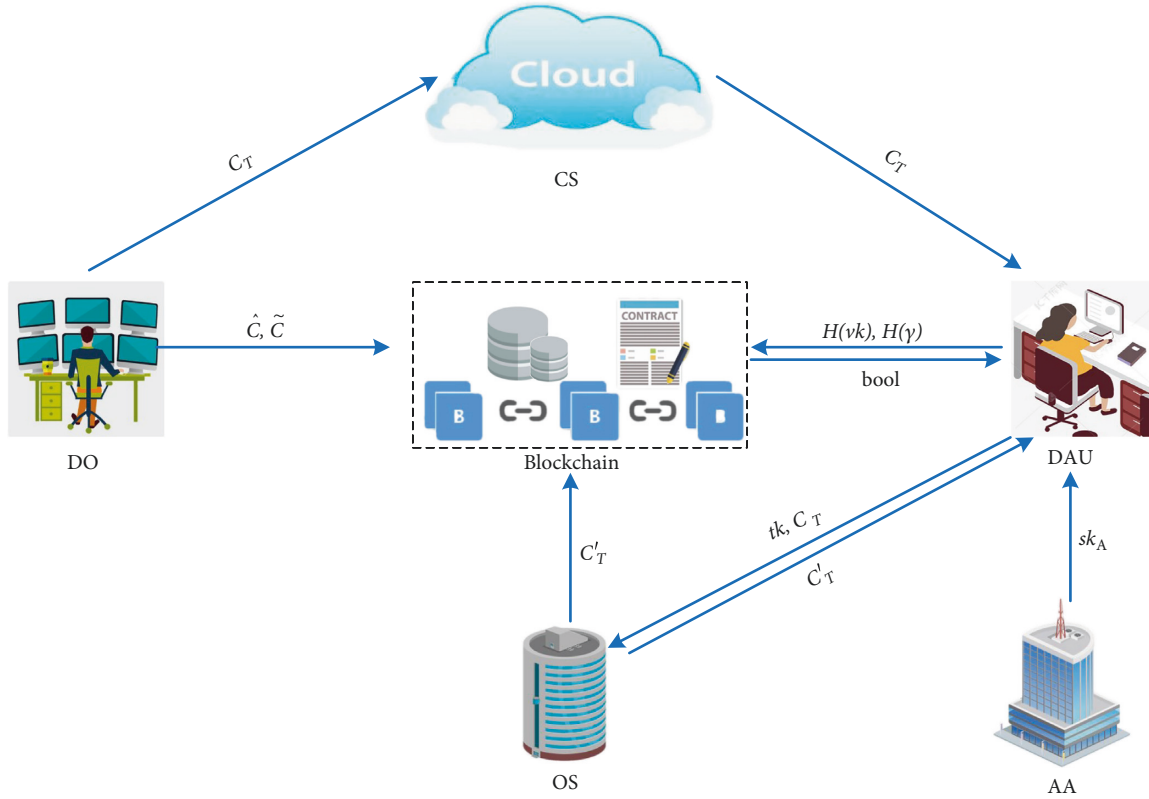


FIGURE 1: System model.

being authorized. Compared with the public blockchain, the consortium blockchain maintains the characteristics of decentralization and enhances the control of the participating members.

A smart contract is an automatic piece of code deployed on the blockchain with a unique address [35]. The initializer can establish a smart contract and save it as a transaction on the blockchain platform. When a transaction in the contract is triggered, the contract will automatically execute predefined content according to the script, such as executing relevant calculations. Finally, the output and status information of the transaction are recorded in the blockchain as transactions. In our structure, we employ smart contracts to create interfaces for the blockchain application layer and verify operations through the interaction of cloud servers with smart contracts instead of using semitrusted servers.

4. Our Cloud Storage Data Access Control Scheme Based on Blockchain and Attribute-Based Encryption

4.1. System Model. Figure 1 depicts the framework of our data access control system, which includes six entities: Attribute Authority, Data Owner, Cloud Server, Data Accessing Users, Blockchain, and Outsourcing Server. The functions of various entities are described as follows:

- (i) The *Attribute Authority (AA)* is responsible for setting up the system and generating the users' private keys.

- (ii) The *Data Owner (DO)* calculates the hash of the initial data and parameters used for authentication and uploads these components to the blockchain platform. Then, the DO generates the ciphertext by encrypting the plaintext according to the access policy and sends it to the cloud server for storage.
- (iii) The *Cloud Server (CS)* is a semitrusted entity that stores data ciphertext.
- (iv) The *Data Accessing User (DAU)* is initially involved in generating a key that is used by the outsourcing server for decryption. After receiving the storage address returned by the cloud server, the DAU is responsible for computing parameters and decrypting. After obtaining the plaintext, the DAU verifies the integrity of the data through the computation.
- (v) *Blockchain.* We use a consortium blockchain with smart contracts deployed. The blockchain platform is responsible for storing verification components and smart contracts, ensuring the correctness of the outsourcing decryption result.
- (vi) The *Outsource Server (OS)* is responsible for detecting the attributes of the accessing user and obtaining the semiciphertext through decryption.

4.2. Security Model. To fulfil the confidentiality and verifiability of the proposed scheme, we define the security model of our scheme by the following two security games.

Game 1 (confidentiality): for our scheme, we define an indistinguishable game under the chosen-ciphertext attack (IND-CCA) that includes an adversary Algorithm A and a challenge Algorithm B .

Initialization phase: B runs Setup(1^λ) to produce the system public key pk and the system master private key msk . Then, B sends pk to A and retains msk .

Inquiry phase 1: A adaptively asks B for the private key of the attribute set Λ , and the private key can be requested repeatedly. B runs KeyGen(pk, msk, Λ) and returns sk_Λ to A .

Challenge phase: A sends equal-length messages M_0 and M_1 as well as access structures W_0 and W_1 to B . B selects $\xi \in \{0, 1\}$ and runs Encrypt(pk, m, W) to generate challenge ciphertext C^* . Finally, B sends C^* to A .

Inquiry phase 2: this is similar to inquiry phase 1, but A cannot ask for the messages M_0 and M_1 .

Guess: A outputs the guess $\xi' \in \{0, 1\}$ of the challenge ciphertext C^* . If $\xi = \xi'$, then B outputs 1, which means that A wins Game 1 with a probability of $\text{Adv} = |\Pr[\xi = \xi'] - 1/2|$.

Theorem 1. *If there is no polynomial-time adversary to attack the above security model with a nonnegligible probability advantage, then our proposed scheme is IND-CCA.*

Game 2 (verifiable): We use the interactive game between adversary F and challenger C to prove the verifiability of our scheme supporting the hidden strategy. The process is as follows:

Initialization phase: C runs the Setup algorithm to produce the master key msk and the system public key pk , while pk is sent to F .

Challenge phase: F asks for the decryption key by specifying an arbitrary set of attributes Λ to be sent to C for inquiry. Then, C performs a key generation algorithm based on the attribute set Λ to generate a decryption key sk . Finally, sk is returned to adversary F .

Output phase: F outputs an access structure W that satisfies the attribute set Λ and a tuple $(C'_T, tk, C_{k1}, C_{k2}, \Delta)$. C executes the preauthentication algorithm to obtain the session key nk_1, nk_2 . If $nk_1 \neq nk_2$, then we claim that F wins the game. We define $\text{Pr}[F \text{ wins}]$ to denote the advantage of F winning the game.

Theorem 2. *If there is a polynomial adversary F who can win the above interactive game with the advantage $\text{Pr}[F \text{ wins}]$, then our attribute-based encryption scheme with the hidden strategy can be considered to be verifiable.*

4.3. Scheme Framework. The operational flow of the cloud storage data access control scheme based on blockchain and

attribute-based encryption is shown in Figure 2, and the specific implementation details of this scheme are as follows.

4.4. Scheme Construction

4.4.1. System Setup. The credible attribute authorization centre (AA) executes the system setup algorithm. is a group generation algorithm that outputs tuple $(N = p_1 p_2 p_3, G, G_T, \hat{e})$. AA first selects a security parameter λ and runs the algorithm $\varphi(\lambda)$ to obtain the system parameters $(N = p_1 p_2 p_3, G_0, G_T, \hat{e})$, where G_0 and G_T are two cyclic groups of order N , and p_1, p_2 , and p_3 are three different prime numbers. G_{p_1}, G_{p_2} , and G_{p_3} are three subgroups from G_0 , whose generators are g_1, g_2 , and g_3 , respectively. We suppose that $U = \{\text{att}_1, \text{att}_2, \dots, \text{att}_n\}$ is a system attribute set and $S_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,j}\}$ is the value set of the attribute att_i . For any attribute att_i in the system, AA generates a public key pk and a master key msk according to the following steps:

- (1) AA chooses two hash functions in cryptography $H: \{0, 1\}^* \rightarrow Z_N$ and $H_0: G_0 \rightarrow Z_N^*$, which are anticollision.
- (2) For any attribute att_i in the system, AA randomly selects $x_{i,j} \in Z_N^*$ and $Q_{i,j} \in G_{p_3}$ and calculates $A_{i,j} = g_1^{1/x_{i,j}} Q_{i,j}$, where $i \in (1, 2, \dots, n), j \in (1, 2, \dots, n_i)$.
- (3) AA randomly selects $\beta_0, \beta \in Z_N^*$ and $Q_0 \in G_{p_3}$ and then calculates $Y_0 = e(g_1, g_1)^{\beta_0}$ and $Y = e(g_1, g_1)^\beta$.
- (4) AA defines a key distribution function KF that maps the session key to a stream of bits of length κ and two parameters ω and ν that belong to G_{p_3} .
- (5) AA publishes the public key $pk = (A_0, g_3, \{A_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq n_i}, Y_0, Y, KF, \omega, \nu, \kappa, H, H_0)$ and keeps the master private key $msk = (g_1, \{x_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq n_i}, \beta_0, \beta)$ secretly.

4.4.2. Key Generation. According to the attribute list Λ of DAU, AA randomly selects $\lambda_i \in Z_N^*$ for any attribute $i (1 \leq i \leq k)$ and calculates $K_0 = g_1^{\beta_0 - \sum_{i=1}^k \lambda_i}$, $K = g_1^{\beta - \sum_{i=1}^k \lambda_i}$ and $K_i = g_1^{\lambda_i x_{i,j}}$. Then, AA sends the generated private key $sk_\Lambda = (K_0, K, \{K_i\}_{1 \leq i \leq k})$ to DAU.

4.4.3. Verification Component Generation. The data owner (DO) performs the following operations to generate and upload verification components.

- (1) The DO randomly selects $s \in Z_N$ and a session key $nk = Y_0^s = e(g_1, g_1)^{\beta_0 s}$ and uses the key distribution function $KF(nk, \kappa) = vk \parallel \gamma$ defined by AA, where γ is a random value and vk is the verification key. Then, the DO calculates $\hat{C} = \omega^{H(vk)} \gamma^{H(\gamma)}$, which is used to verify the outsourcing decryption result.

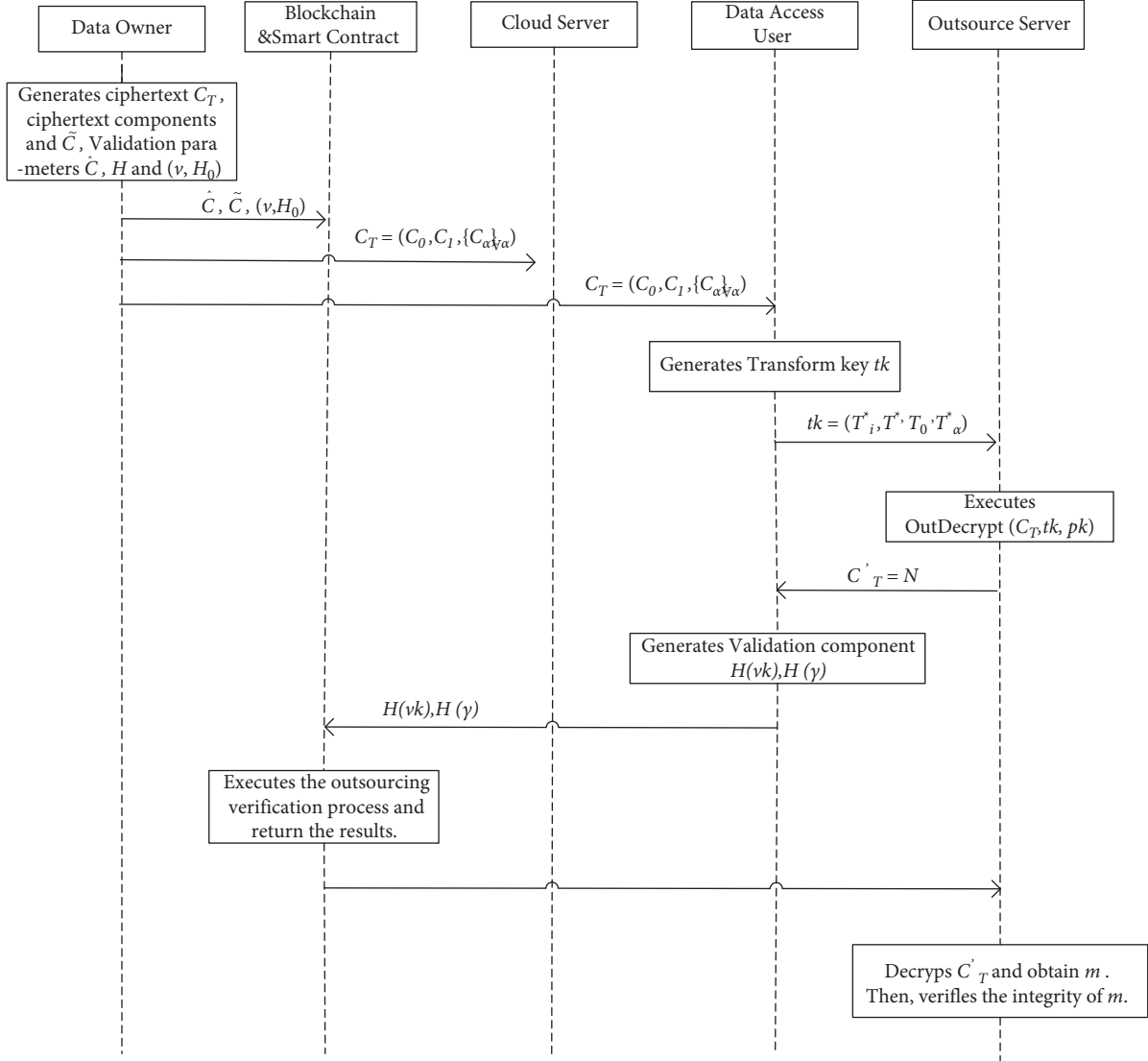


FIGURE 2: Framework of the proposed system.

- (2) The DO computes $\tilde{C} = v^{H_0(m)}$ and uploads to the blockchain platform. The stored addresses Add_m and (v, H_0) are sent to the smart contract as verification components.

4.4.4. Data Encryption. We adopt the access structure used in Zhao et al. scheme [30]. The DO performs the following operations with the access policy W to encrypt plaintext M .

- (1) The DO selects a random element $Q'_0 \in G_{p_3}$ and then calculates $C_0 = A_0^s Q'_0$ and $C_1 = mY^s$.
- (2) The DO sets the secret value s as the root node's value of the access tree. Then, the status of leaf nodes is set to read. Apart from leaf nodes, the status of all child nodes is set to unread. Later, the DO performs a recursive operation for each node with an unread state:

- (a) If the nonleaf node represents a logic "AND," then DO sets s_i for the $u - 1$ previous nodes of its children. Then, the value of the last leaf node is calculated by $s_u = s - \sum_{i=1}^{u-1} s_i$.
- (b) If the nonleaf node delegates a logic "OR," then DO sets s as the value of all child nodes, while the state of these nodes is set to read.
- (c) If the nonleaf node expresses the "threshold" with a threshold value h , then the DO randomly generates a polynomial f of degree $h - 1$. Meanwhile, the polynomial satisfies $f(0) = s$ and assigns the value of $f(i)$ to the i th child node.

- (3) The DO enforces operations to hide the policy. For simplicity, the parent node of any leaf node is named PNode. Suppose a PNode α exists, which is assigned the secret value s_α . Γ_α represents a subtree in which α is the root node, and all leaf nodes are indicated by a set S_{Γ_α} . For each attribute att_i , DO calculates $C_{i,j}$

according to different conditions. When an attribute $att_i \in \Gamma_\alpha$ and the value $v_{i,j} \notin S_{\Gamma_\alpha}$, the DO randomly selects $s_{i,j} \in Z_N^*$ and $Q'_{i,j} \in G_{p_3}$ and calculates $C_{i,j} = A_{i,j}^{s_{i,j}} Q'_{i,j}$. Otherwise, DO calculates $C_{i,j} = A_{i,j}^{s_{i,j}} Q'_{i,j}$.

- (4) The DO randomly selects $Q_\alpha \in G_{p_3}$, calculates $\bar{C}_\alpha = A_0^{s_\alpha} Q_\alpha$ and $I_\alpha = Y^{s_\alpha}$ for each PNode α , and obtains the component of the ciphertext $C_\alpha = (\bar{C}_\alpha, I_\alpha, \{C_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq n_i})$.
- (5) The DO obtains the entire ciphertext $C_T = (C_0, C_1, \{C_\alpha\}_{\forall \alpha})$ and sends it to the CS for storage.

4.4.5. Transformation Key Generation. DAU randomly chooses a factor $y \in Z_N^*$ and calculates $T_i^* = T_i^{1/y}$, $T^* = T^{1/y}$, $T_0^* = T_0^{1/y}$, and $I_\alpha^* = I_\alpha^{1/y}$. Later, DAU sends the transformation key $tk = (T_i^*, T^*, T_0^*, I_\alpha^*)$ and semidecrypted ciphertext C_T' to the outsourcing server OS.

$$\text{PreDecNode}(\beta) = \begin{cases} \prod_{i=1}^u \text{PreDecNode}(\text{child}(\beta, i)), & \text{structure}(\beta) = \text{AND}, \\ \text{PreDecNode}(\text{child}(\beta, i)), & \text{structure}(\beta) = \text{OR}, \\ \prod_{i=1}^h \text{PreDecNode}(\text{child}(\beta, i))^{\Delta_{i,\beta_0}}, & \text{structure}(\beta) = \text{Threshold}. \end{cases} \quad (1)$$

Finally, OS calculates $\omega_\chi = I_\alpha^* / e(C_\alpha, T^*) \text{PreDecNode}(\alpha)$.

- (2) Only when $\omega_\chi = 1$, does the OS further calculate $N = e(C_0, T_0^*) \text{PreDecNode}(\text{root}(W))$ in the decryption phrase. Then, the OS sends the semidecrypted ciphertext $C_T' = N$ to the DAU.

The preauthentication DAU obtains the semidecrypted ciphertext C_T' and generates the computed values $H(vk)$ and $H(\gamma)$ to complete the preauthentication work.

- (1) DAU uses the blinding factor γ and computes the session key $nk = N^{-\gamma} = e(g_1, g_1)^{s\beta_0}$.
- (2) DAU executes $KF(nk, \kappa) = vk \parallel \gamma$ mapping the session key to a stream of bits of length κ . Finally, the DAU calculates $H(vk)$ and $H(\gamma)$ sends it to the smart contract.

4.4.7. Outsourcing Verification. Receiving the elements $H(vk)$ and $H(\gamma)$ from the DAU, the smart contract computes $\omega^{H(vk)} \gamma^{H(\gamma)}$. If equation $\omega^{H(vk)} \gamma^{H(\gamma)} = \hat{C}$ holds, then the smart contract outputs $\text{bool} = 1$. Otherwise, the algorithm is terminated.

4.4.8. Decryption and Integrity Verification. If DAU receives $\text{bool} = 1$, then the semidecrypted ciphertext C_T' computed by the OS is not fake. Then, the steps of decryption and verification by the DAU are as follows:

- (1) DAU utilizes γ to compute plaintext $m = \hat{C} \cdot N^\gamma$.

4.4.6. Outsourcing Decryption. Execution by the outsourcing server OS. The algorithm is divided into an attribute detection phase and a decryption phase. The attribute detection phase is to preeliminate the attribute values in the private key that are unable to meet the access policy. This design can avoid bottom-up recursive decryption to reduce computational overhead. Only after passing the attribute checking can the algorithm proceed to the decryption phase.

- (1) The OS runs different functions according to different nodes in the access structure to detect the value. If a node is PNode α , the OS runs $\text{PreDecNode}(\alpha) = \prod_{i=1}^k e(C_{i,j}, T_i^*)$. Likewise, if a node is a normal node β , according to the structure of "OR", "AND" and "Threshold" in the access structure, then the OS runs $\text{PreDecNode}(\beta)$.

- (2) DAU computes $\tilde{C}' = v^{H_0(m)}$ and determines whether the computed \tilde{C}' equals \tilde{C} . If equation $\tilde{C}' = \tilde{C}$ holds, then the ciphertext stored on the cloud is proved completely.

5. Correctness Analysis

5.1. Correctness of Data Decryption. Here, we verify the correctness of the outsourcing decryption algorithm (executed by OS) and decryption algorithm (by DAU).

Receiving $tk = (T_i^*, T^*, T_0^*, I_\alpha^*)$ sent from the user, the OS executes attribute detection. The OS judges whether the user access structure satisfies all s_α values through the ω_χ result value calculated in the attribute detection phrase. The calculation equation is as follows:

$$\begin{aligned} \omega_\chi &= \frac{I_\alpha^*}{e(C_\alpha, T^*) \text{PreDecNode}(\alpha)} \\ &= \frac{\gamma^{s_\alpha/y}}{e(A_0^{s_\alpha} Q_\alpha, g_1^{\beta-\lambda/y}) e(g_1, g_1)^{\lambda s_\alpha/y}}, \\ &= \frac{e(g_1, g_1)^{\beta s_\alpha/y}}{e(A_0^{s_\alpha} Q_\alpha, g_1^{\beta-\lambda/y}) e(g_1, g_1)^{\lambda s_\alpha/y}} \\ &= \frac{e(g_1, g_1)^{\beta s_\alpha/y}}{e(g_1, g_1)^{s_\alpha(\beta-\lambda)/y} e(g_1, g_1)^{\lambda s_\alpha/y}} = 1. \end{aligned} \quad (2)$$

Only when the user's attributes pass the detection, can the OS obtain $\omega_\chi = 1$; otherwise, ω_χ is a random value. After

receiving $\omega_\chi = 1$, the OS uses tk to calculate C'_T , and the calculation equation is as follows:

$$\begin{aligned}
N &= e(C_0, T_0^*) \text{Pre Dec Node}(\text{root}(W)) \\
&= e(A_0^s Q_0', T_0^{1/y}) \text{Pre Dec Node}(\text{root}(W)) \\
&= e(g_0^s Q_0^s Q_0', g_1^{\beta_0 - \lambda/y}) e(g_1, g_1)^{\lambda s/y} \\
&= e(g_1, g_1)^{s(\beta_0 - \lambda)/y} e(g_1, g_1)^{\lambda s/y} \\
&= e(g_1, g_1)^{s\beta_0/y}.
\end{aligned} \tag{3}$$

DAU receives the C'_T sent from the OS and then calculates $nk = N^{-y} = e(g_1, g_1)^{s\beta_0}$ and $KF(nk, \kappa) = vk\|\gamma$. The smart contract verifies whether semidecrypted ciphertext C'_T is valid. If equation $\omega^{H(vk)\gamma^{H(\gamma)}} = \tilde{C}$ holds, then the decryption result from the OS is correct. Then, DAU using N , \tilde{C} and y recover the plaintext by the following:

$$\begin{aligned}
\tilde{C} \cdot N^y &= \frac{mY^s}{e(g_1, g_1)^{s\beta_0}} \\
&= \frac{me(g_1, g_1)^{s\beta_0}}{e(g_1, g_1)^{s\beta_0}} = m.
\end{aligned} \tag{4}$$

5.2. Integrity of Cloud Data. After the DAU obtains the plaintext, he or she calculates $\tilde{C}' = v^{H_0(m)}$ and verifies that \tilde{C}' is equal to the \tilde{C} stored on the blockchain. If $\tilde{C}' \neq \tilde{C}$, then the tampering of the ciphertext by the cloud server is demonstrated.

6. Security Analysis

6.1. Confidentiality. Data confidentiality of our scheme relies on the security of the attribute encryption system. This section proves Theorem 1 based on the security model in Section 4.2.

Theorem 3. *If there is no polynomial-time adversary that can attack the scheme of [30] with a nonnegligible advantage, then no polynomial adversary A can break the scheme of this article with a nonnegligible advantage.*

Proof. Based on the proof method in Scheme [30], we prove that the confidentiality of our scheme satisfies security under a chosen-ciphertext attack.

The following simulation game is played between adversary A and challenger B .

Initialization phase: B runs $\text{Setup}(1^\lambda)$ to produce the system public key $pk = (A_0, g_3, \{A_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq n_i}, Y_0, Y, KF, \omega, \nu, \kappa, H, H_0) \neq$ and the system master private key $msk = (g_1, \{x_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq n_i}, \beta_0, \beta)$. Then, B sends pk to A and generates an initially empty list L and an empty set \mathbb{R} .

Inquiry phase 1: A can initiate the following two types of inquiries to B .

- (1) Private key inquiry: A adaptively asks B for the private key of the attribute set Λ , B runs $\text{Key Gen}(pk, msk, \Lambda)$ and returns $sk_\Lambda = (K_0, K, \{K_{i,j}\}_{1 \leq i \leq k})$ to A . B calculates $\Lambda \cap \mathbb{R}$ and assigns $\Lambda \cap \mathbb{R}$ to \mathbb{R} .
- (2) Transformation key inquiry: receiving the request of token inquiry from A , B first searches for $(\Lambda, sk_\Lambda, tk_\Lambda)$ in list L . If $(\Lambda, sk_\Lambda, tk_\Lambda)$ exists, then B returns tk_Λ to A ; otherwise, B chooses a random number $y \in Z_N$ and calculates $tk_\Lambda = (T_i^*, T^*, T_0^*, I_\alpha^*)$. Then, B adds Λ and tk_Λ to list L and returns list L to A .

Challenge phase: A sends equal-length messages M_0, M_1 and access structure W_0, W_1 to B . B selects $\gamma \in \{0, 1\}$ and runs $\text{Encrypt}(pk, m, W)$ to generate challenge ciphertext $C_T = (C_0 = A_0^s Q_0', C_1 = mY^s, \{C_\alpha\}_{\forall \alpha})$. Finally, B sends C_T to A .

Inquiry phase 2: similar to inquiry phase 1, but A cannot ask for messages M_0 and M_1 .

Guess: A outputs the guess $\gamma' \in \{0, 1\}$. If $\gamma' = \gamma$, then the attack is declared successful. Based on the proof of Definition 5 in Scheme [30], it is difficult for A to guess γ' and γ selected randomly during the ciphertext generation phase. We prove that the confidentiality of our scheme satisfies security under a chosen-ciphertext attack. \square

6.2. Privacy Policy. The DO uploads the ciphertext components $C_0 = A_0^s Q_0', C_1 = mY^s$ and $C_\alpha = (\bar{C}_\alpha, I_\alpha, \{C_{i,j}\}_{(1 \leq i \leq n, 1 \leq j \leq n_i)})$ to the CS, where $C_{i,j} = A_{i,j}^{s_\alpha} Q'_{i,j}$ or $C_{i,j} = A_{i,j}^{s_{i,j}} Q'_{i,j}$. Note that the attribute information s_α is hidden in the ciphertext component C_α . When an attribute value of the accessing user satisfies the value under node α , then the ciphertext component can be obtained by $C_{i,j} = A_{i,j}^{s_\alpha} Q'_{i,j}$, where s_α is the attribute information. When a data user does not meet the access control, the DO uses a random value $s_{i,j}$ to replace s_α and obtains the ciphertext component $C_{i,j} = A_{i,j}^{s_{i,j}} Q'_{i,j}$, even if the data user who does not meet the access control obtains the ciphertext and calculates

$$\begin{aligned}
\omega_\chi &= \frac{I_\alpha^*}{e(C_\alpha, T^*) \text{Pre Dec Node}(\alpha)} = \frac{Y^{s_\alpha/y}}{e(A_0^s Q_\alpha, g_1^{\beta - \lambda/y}) e(A_{i,j}^{s_{i,j}} Q'_{i,j}, K_i^{1/y})^\lambda} \\
&= \frac{e(g_1, g_1)^{\beta s_\alpha/y}}{e(g_1, g_1)^{s_\alpha(\beta - \lambda)/y} e(g_1, g_1)^{\lambda s_{i,j}/y}}
\end{aligned} \tag{5}$$

There are random values $s_{i,j}$ in the above equation; therefore, users who do not satisfy the access control do not obtain the attribute values of node α . Thus, the whole access structure cannot be inferred from the access policy. Therefore, the scheme in this article satisfies policy privacy.

TABLE 1: Functional comparison.

System	Access structure	Hidden policy	Predicted decryption	Verifiable outsourcing	Integrity	Confidentiality	Blockchain technology
[36]	LSSS	√	×	×	×	CPA	×
[30]	Access tree	√	√	√	×	CPA	×
[34]	LSSS	√	×	√	√	—	√
[29]	LSSS	√	×	√	×	CPA	×
Ours	Access tree	√	√	√	√	CCA	√

6.3. Verifiability

Theorem 4. *For a composite-order bilinear group, if the discrete logarithm problem holds in the system, then the proposed scheme satisfies verifiability.*

Proof. If within the PPT time, the verifiability of the system can be attacked by attacker A with a nonnegligible advantage, then algorithm \mathbb{S} can be simulated to solve the discrete logarithm problem in a composite-order bilinear group system. The bilinear system $(N, p_1, G_0, G_T, \hat{e}, g_1, \Delta = g_1^x)$ is input into the simulation algorithm \mathbb{S} . The algorithm \mathbb{S} needs to calculate $x = \log_g \Delta$. The game process between the simulation algorithm \mathbb{S} and attacker A is as follows:

Initialization phrase: the simulation algorithm \mathbb{S} randomly generates the parameters $\gamma \in Z_N$, picks two anticollision hash functions, $H: \{0, 1\}^* \rightarrow Z_N$ and $H_0: G_0 \rightarrow Z_N^*$, and defines a key distribution function KF . Later, \mathbb{S} generates system public parameters $pk = (A_0, g_3, \{A_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq n_i}, Y_0, Y, KF, \omega, \nu, \kappa, H, H_0)$ according to the scheme initialization process and sends the public key to attacker A .

Challenge phrase: Attacker A sends the attribute set Λ to the simulation algorithm \mathbb{S} , performs the key generation process $\text{Key Gen}(pk, \text{msk}, \Lambda)$ to generate the private key $sk_\Lambda = (K_0, K, \{K_i\}_{1 \leq i \leq k})$ corresponding to the attribute set Λ and sends it to attacker A .

Output phrase: Attacker A outputs a tuple $(C_T', tk, C_{k_1}, C_{k_2}, \Delta)$ and an encrypted access structure W that satisfies the attribute set Λ . The simulation algorithm \mathbb{S} calculates $KF(nk_1, \kappa) = vk_1 \parallel \varepsilon_1$ and $KF(nk_2, \kappa) = vk_2 \parallel \varepsilon_2$, where $nk_1 = w_{1,2}^\Delta$ and $nk_2 = w_{2,2}^\Delta$. If $nk_1 \neq nk_2$, that is, attacker A wins the game, and the simulation algorithm \mathbb{S} calculates

$$\begin{aligned}
g_1^{x \cdot H(vk_1) + \gamma \cdot H(\varepsilon_1)} &= \omega^{H(vk_1) \nu^H(\varepsilon_1)} \\
&= w_1 \\
&= g_1^{x \cdot H(vk_2) + \gamma \cdot H(\varepsilon_2)} \\
&= \omega^{H(vk_2) \nu^H(\varepsilon_2)}.
\end{aligned} \tag{6}$$

Because the selected hash function H has collision resistance, $vk_1 \neq vk_2$ and $H(vk_1) \neq H(vk_2)$, the algorithm \mathbb{S} is able to compute $x = \gamma(H(\varepsilon_1) - H(\varepsilon_2)) / (H(vk_1) - H(vk_2))$ as

a solution to the discrete logarithm problem, which proves that the proposed scheme is verifiable. \square

6.4. Data Integrity. Data integrity is guaranteed by two processes. First, the smart contract is used to realize the decryption correctness of the outsourcing server. Subsequently, the original data hash on the blockchain is saved to verify the data integrity. After receiving the semidecrypted ciphertext $C_T' = N$ sent by the outsourcing server, the data access user uses the blinding factor γ to calculate the session key $nk = N^{-\gamma} = e(g_1, g_1)^{\beta \gamma}$ and replaces the key allocation function $KF(nk, \kappa) = vk \parallel \gamma$. A smart contract verifies equation $\omega^{H(vk) \nu^H(\gamma)} = \tilde{C}$ and outputs $\text{bool} = 1$ when this equation is established. Then, the data access user continues to decrypt semidecrypted ciphertext. Otherwise, the smart contract outputs $\text{bool} = 0$ and ends the decryption.

After the DAU performs decryption to obtain plaintext m , $\tilde{C}' = \nu^{H_0(m)}$ is calculated and the validity of $\tilde{C}' = \tilde{C}$ is verified. If the equation does not hold, then it cannot be verified by data integrity.

7. Performance Analysis

7.1. Property Analysis. In this section, the functionality of our system is compared with schemes in [29, 30, 34, 36], and the comparison outcomes are shown in Table 1. We can note from Table 1 that our scheme is the only one that meets the requirements of policy hiding, verifiable outsourcing, and data integrity under CCA. Schemes in [29, 34, 36] use outsourcing for decryption operations, but their decryption operations are not very efficient. Moreover, the scheme in [36] does not support the validation of outsourcing decryption results. In addition, schemes in [29, 30, 36] achieve data integrity verification by relying on a trusted cloud server. As a result, the proposed new scheme is able to provide both higher security and fuller functionality than existing similar schemes.

7.2. Performance Evaluation. We compare our scheme with Systems [30, 34, 36], which also use bilinear groups of composite order. The computational cost of these schemes is analysed through three stages: encryption, decryption, and outsourcing decryption, and the comparison results are shown in Table 2. Our scheme mainly considers pair operations and exponential operations in groups G and G_T . We use G and G_T to denote the time to perform an exponential

TABLE 2: Computational overhead comparison.

System	Encryption	Outsourcing decryption Predetected	User decryption (s)	Decryption
[30]	$(1 + n_a + n_a n)G + (1 + n)G_{T_p}$	$(m + n)(T_p + G_T)$	$T_p + G_T$	$2G_T = 0.42$
[34]	$T_p + (3 + 3n_l)G + G_T$	$(2 + n_l)T_p + (1 + 2n_l)G$	0	
[36]	$n_l T_p + (4 + 3n_l)G + G_T$	$n_k T_p + (n_k + 1)G$	$G = 0.72$	
Ours	$(n_a + n_a n)G + nG_T$	$(m + n)(T_p + G_T)$	$T_p + G_T$	$G_T = 0.21$

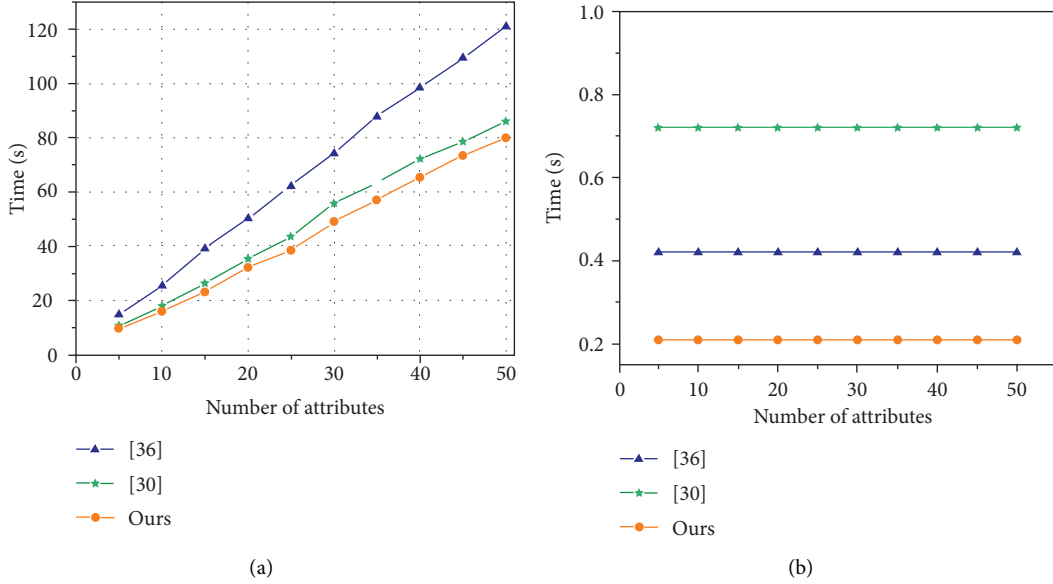


FIGURE 3: Time cost of encryption and decryption with different numbers of attributes. (a) Encryption time of data owner. (b) Decryption time of the user side.

operation on the corresponding group and T_p to denote the time to perform a logarithmic operation. Furthermore, the number of authorized attributes in the system is denoted by n_w , the number of leaf node parents by n_a , the number of attributes in the key by n_k , and the number of user attributes by n_l .

To evaluate the specific computational performance of our scheme, we conducted experiments. Our experimental environment is an Intel(R) Core (TM) i5-8250U CPU 1.80 GHz processor with 8 GB memory and the Win10 operating system (Pairing-Based Cryptography, PBC) library in the VC6.0 environment. Through the above environment, the new scheme was simulated and compared with schemes in [30, 36], and the experimental data were averaged over 20 runs. In the composite-order bilinear group, the times of G , G_T , and T_p are 0.21 s, 0.72 ms, and 1.64 s, respectively. Our scheme and Zhao et al. proposed a scheme in [30] that adopts a special access number structure, and the encryption time is related to the number of parent nodes of leaf nodes n_a . As a result, to better reflect the two systems' performance, we set $n_a = 1$. In addition, we suppose the user has 5 attributes. The number of attributes connected with ciphertext is half the number of systems, and the system contains between 5 and 50 attributes.

In Table 2, we compare these schemes in terms of computational overhead, mainly considering the cost of encryption, outsourcing decryption, and user decryption. For encryption, our scheme improves the efficiency of the ciphertext generation stage. Unlike the scheme in [30], the new scheme uses blockchain technology and minimizes the number of ciphertext components that must be uploaded to the cloud server. Consequently, two exponential operations originally performed by the data owner in the encryption process are reduced. Additionally, in the correctness verification process, the new scheme leaves the verification of the outsourcing results to be performed by smart contracts, reducing the verification overhead for local users. In the decryption phase, all four experiments presented in Table 2 use an outsourced server for predecryption so the decryption overhead for the user is kept at a constant level. The calculation times of the three schemes are G , $2G$, and G_T . Compared with the scheme in [34] without local overhead, and although the new scheme has some decryption overhead, its security is better than the scheme in [34]. On the one hand, when the scheme in [34] uses smart contracts to verify the results of outsourcing, it needs to know the blinding factor that is private for the user. On

the other hand, the smart contract decrypts and obtains the plaintext instead of the user, which makes the plaintext information available to the smart contract and increases the risk of data leakage.

Figure 3 shows the time taken to perform the operation of the data owner and user side. We experiment with different attribute values and show the encryption time changes of the new scheme, the scheme in [30], and the scheme in [34], in Figure 3(a). The computational overhead of the new scheme and the scheme in [30] is smaller than that of the scheme in [34], as shown in Figure 3(a), and the advantage grows as the number of characteristics grows. Due to the additional pair operations and exponential operations in group G that must be computed while hiding the access control policy, the scheme in [34] takes longer. Moreover, based on the scheme in [30], our scheme introduces blockchain technology to encrypt the ciphertext components that need to be encrypted with a data owner in advance in their scheme. This design reduces the encryption time of two exponential operations in the ciphertext generation process.

From Figure 3(b), we can clearly see that the attributes are irrelevant to the time taken for the three schemes to perform decryption (user side) operations, but the time expenditure advantage of our scheme is always higher than those of Schemes [30, 36].

8. Conclusion

We propose a verifiable access control model for outsourced cloud storage that supports policy hiding as well as secure and efficient decryption. Our system is based on the CP-ABE, avoiding privacy leakage by hiding access policies. The idea of outsourcing and a more efficient decryption algorithm reduce the computational cost of local users and outsourcing decryption servers in the decryption process, respectively. To validate the integrity of outsourced decryption results, we use smart contracts implemented on the blockchain, which implements a decentralized ciphertext result verification approach. At the same time, through the hash of the original data retained on the blockchain platform, the integrity of the decrypted data is verified, which solves the dependence of the traditional scheme on fully trusted cloud servers. The analysis results show that the new scheme not only improves computing performance and meets CCA security but also verifies data integrity in the cloud storage environment. In future work, we will attempt to improve the cloud storage data access control scheme for multi-authorization centres.

Data Availability

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

Conflicts of Interest

All authors have no conflicts of interest.

Acknowledgments

This research was supported by the China Postdoctoral Science Foundation (no. 2017M610817) and the Gansu Science and Technology Planning Project (no. 20CX9ZA076).

References

- [1] H. Yang, Z. Yi, R. Li et al., "Improved Outsourced Provable Data Possession for Secure Cloud Storage," *Security and Communication Networks*, vol. 2021, Article ID 1805615, 12 pages, 2021.
- [2] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-based big data storage systems in cloud computing: perspectives and challenges," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 75–87, 2017.
- [3] W. B. Kim, D. Seo, D. Kim, and I.-Y. Lee, "Group Delegated ID-Based Proxy Reencryption for the Enterprise IoT-Cloud Storage Environment," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 7641389, 12 pages, 2021.
- [4] S. Qi, Y. Lu, W. Wei, and X. Chen, "Efficient data access control with fine-grained data protection in cloud-assisted IIoT," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2886–2899, 2021.
- [5] M. Joshi, K. Joshi, and T. Finin, "Attribute Based Encryption for Secure Access to Cloud Based EHR Systems," in *Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pp. 932–935, San Francisco, CA, USA, July 2018.
- [6] R. Walid, K. P. Joshi, S. Geol Choi, and D.-y. Kim, "Cloud-based Encrypted EHR System with Semantically Rich Access Control and Searchable Encryption," in *Proceedings of the 2020 IEEE International Conference On Big Data (Big Data)*, pp. 4075–4082, Atlanta, GA, USA, December 2020.
- [7] S. Li, Q. Zhang, X. Wu, W. Han, and Z. Tian, "Attribution classification method of APT malware in IoT using machine learning techniques," *Security and Communication Networks*, vol. 2021, Article ID 9396141, 12 pages, 2021.
- [8] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, vol. 37, 2021.
- [9] Y. Chen, S. Dong, T. Li, Y. Wang, and H. Zhou, "Dynamic multi-key FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5239–5249, 2021.
- [10] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pp. 321–334, Berkeley, CA, USA, May 2007.
- [11] B. Waters, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Public Key Cryptography - PKC 2011*, Springer, Berlin, Germany, 2011.
- [12] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008, <http://bitcoin.org/bitcoin.pdf>.
- [13] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, pp. 3596–3612, 2021.
- [14] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, 2021.

- [15] A. Sahai, "Fuzzy identity-based encryption," in *Lecture Notes in Computer Science*, B. Waters, Ed., Springer, Berlin, Germany, pp. 457–473, 2005.
- [16] G. Lin, H. Hong, and Z. Sun, "A collaborative key management protocol in ciphertext policy attribute-based encryption for cloud data sharing," *IEEE Access*, vol. 5, pp. 9464–9475, 2017.
- [17] C. Li, J. He, L. Cheng, C. Guo, and K. Zhou, "Achieving privacy-preserving CP-ABE access control with multi-cloud," in *Proceedings of the 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/Sustain-Com)*, pp. 801–808, Melbourne, Australia, December 2018.
- [18] A. Kapadia, P. P. Tsang, and W. S. Smith, "Attribute-based Publishing with Hidden Credentials and Hidden Policies," in *Proceedings of the Network And Distributed System Security Symposium*, pp. 179–192, NDSS 2007, San Diego, CA, USA, February 2007.
- [19] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based Encryption with Partially Hidden Encryptor-Specified Access Structures," in *Proceedings of the International Conference on Applied Cryptography and Network Security*, pp. 111–129, Springer, Berlin, Heidelberg, June 2008.
- [20] J. Lai, R. H. Deng, and Y. Li, "Fully Secure Ciphertext-Policy Hiding CP-ABE," in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 24–39, Springer, Berlin, Heidelberg, May 2011.
- [21] J. Hur, "Attribute-based secure data sharing with hidden policies in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2171–2180, 2013.
- [22] N. Helil and K. Rahman, "CP-ABE Access Control Scheme for Sensitive Data Set Constraint with Hidden Access Policy and Constraint Policy," *Security and Communication Networks*, vol. 2017, Article ID 2713595, 13 pages, 2017.
- [23] Y. Song, H. Zhen, F. Liu, and L. Liu, "Attribute-based encryption with hidden policies in the access tree," *Journal on Communications*, vol. 36, no. 9, pp. 119–126, 2015.
- [24] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, "Efficient and Provable Secure Ciphertext-Policy Attribute-Based Encryption Schemes," in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 1–12, Springer, Berlin, Heidelberg, April 2009.
- [25] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of abe ciphertexts," in *Proceedings of the USENIX Security Symposium*, San Francisco, CA, USA, August 2011.
- [26] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1343–1354, 2013.
- [27] J. Li, F. Sha, Y. Zhang, X. Huang, and J. Shen, "Verifiable Outsourced Decryption of Attribute-Based Encryption with Constant Ciphertext Length," *Security and Communication Networks*, vol. 2017, Article ID 3596205, 11 pages, 2017.
- [28] J. Zhang, Z. Cheng, X. Cheng, and B. Chen, "OAC-HAS: outsourced access control with hidden access structures in fog-enhanced IoT systems," *Connection Science*, vol. 33, no. 4, pp. 1060–1076, 2021.
- [29] S. Lin, R. Zhang, H. Ma, and M. Wang, "Revisiting attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2119–2130, 2015.
- [30] Y. Zhao, X. Zhang, X. Xie, and S. Kumar, "A verifiable hidden policy CP-ABE with decryption testing scheme and its application in VANET," *Transactions on Emerging Telecommunications Technologies*, p. e3785, 2019.
- [31] D. Di Francesco Maesa, P. Mori, and L. Ricci, "A blockchain based approach for the definition of auditable access control systems," *Computers & Security*, vol. 84, pp. 93–119, 2019.
- [32] Y. Rahulamathavan, R. C.-W. Phan, M. Rajarajan, S. Misra, and A. Kondo, "Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption," in *Proceedings of the 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, Bhubaneswar, India, December 2017.
- [33] Y. Zhang, D. He, and K.-K. R. Choo, "BaDS: blockchain-based architecture for data sharing with ABS and CP-ABE in IoT," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 2783658, 9 pages, 2018.
- [34] F. Zhang, "Research on access control of internet of things based on blockchain and attribute based encryption," Master's Thesis, Nanjing University of Posts and Telecommunications, 2020.
- [35] J. Zhu, K. Hu, and B. Zhang, "Review on formal verification of smart contract," *Acta Electronica Sinica*, vol. 49, no. 4, pp. 792–804, 2021.
- [36] B. Wang and H. Wang, "Research on cloud storage scheme based on attribute encryption," *Journal of Electronics and Information Technology*, vol. 38, no. 11, pp. 2931–2939, 2016.

Research Article

Multiparty Data Publishing via Blockchain and Differential Privacy

Zhen Gu ¹, Kejia Zhang ^{1,2} and Guoyin Zhang ¹

¹College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

²School of Mathematical Science, Heilongjiang University, Harbin 150080, China

Correspondence should be addressed to Kejia Zhang; zhangkejia@hlju.edu.cn

Received 21 January 2022; Accepted 30 March 2022; Published 9 May 2022

Academic Editor: Yuling Chen

Copyright © 2022 Zhen Gu 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.

Data are distributed between different parties. Collecting data from multiple parties for analysis and mining will serve people better. However, it also brings unprecedented privacy threats to the participants. Therefore, safe and reliable data publishing among multiple data owners is an urgent problem to be solved. We mainly study the problem of privacy protection in data publishing. For a centralized scenario, we propose the LDA-DP algorithm. First, the within-class mean vectors and the pooled within-class scatter matrix are perturbed by the Gaussian noise. Second, the optimal projection direction vector with differential privacy is obtained by the Fisher criterion. Finally, the low-dimensional projection data of the original data are obtained. For distributed scenarios, we propose the Mul-LDA-DP algorithm based on a blockchain and differential privacy technology. First, the within-class mean vectors and within-class scatter matrices of local data are perturbed by the Gaussian noise and uploaded to the blockchain network. Second, the projection direction vector is calculated in the blockchain network and returned to the data owner. Finally, the data owner uses the projection direction vector to generate low-dimensional projection data of the original data and upload it to the blockchain network for publishing. Furthermore, in a distributed scenario, we propose a correlated noise generation scheme that uses the additivity of the Gaussian distribution to mitigate the effects of noise and can achieve the same noise level as the centralized scenario. We measure the utility of the published data by the SVM misclassification rate. We conduct comparative experiments with similar algorithms on different real data sets. The experimental results show that the data released by the two algorithms can maintain good utility in SVM classification.

1. Introduction

With the development of science and technology, effective data collection and analysis can help people make better decisions in production. For example, analyzing the information of the patient can help doctors improve the accuracy of diagnosis and level of medical services, and analyzing the trajectory data can improve city traffic congestion. The data contain sensitive information and need to be processed for privacy protection before publishing [1, 2]. There have been some studies on privacy preserving data publishing. For example, the k -anonymity privacy protection technology [3], the encryption technology [4, 5], the blockchain technology [6–8], and differential privacy technology [9–11]. Differential privacy has been widely used for privacy

protection in recent years, the principle of differential privacy is to add random noise to data, which makes the attacker unable to distinguish the original input data. Differential privacy can quantitatively measure the degree of privacy protection and can resist attacks from attackers with background knowledge. Privacy preserving data publishing based on differential privacy has become a research hot spot [12–15].

However, in the distributed scenario, data are possessed by multiple data owners. Data from a single data owner may not be sufficient for statistical learning, and aggregating data by a single data owner may not be possible. For example [16], in Table 1, the data are possessed by three data owners. Each row in Table 1 represents the information of an individual, where records 1 to 4 are from data owner 1, records 5 to 8 are

TABLE 1: Aggregated dataset of each data owner.

ID	Age	Job	Gender	hours-per-week	income
1	39	Shopkeeper	Male	40	> 50K
2	55	Lawyer	Male	13	≤ 50K
3	38	Dancer	Male	20	≤ 50K
4	30	Dancer	Male	25	≤ 50K
5	28	Builder	Female	40	> 50K
6	37	Dancer	Female	23	≤ 50K
7	49	Teacher	Female	16	≤ 50K
8	52	Builder	Male	45	> 50K
9	31	Lawyer	Female	50	> 50K
10	42	Builder	Male	40	> 50K

from data owner 2, and records 9 to 10 are from data owner 3. Simply integrating and publishing the data from each data owner will cause a serious privacy leakage. Sharing and exchange of data in a distributed environment requires security guarantees. In order to solve the proposed problem, we make the following contributions:

- (1) We propose two algorithms which are called LDA-DP and Mul-LDA-DP. The LDA-DP algorithm is used for privacy protection of data publishing in centralized scenario, and the Mul-LDA-DP algorithm is used for privacy protection of data publishing in distributed scenario.
- (2) In the distributed scenario, the data owners cooperate with each other to publish a projection data set which satisfies differential privacy. In order to improve the utility of the published data in the distributed scenario, we propose a correlated noise generation scheme that uses the additivity of the Gaussian distribution to mitigate the effects of noise and can achieve the same noise level as the centralized scenario.
- (3) We conduct experiments on different data sets. The experimental results show that the data released by LDA-DP and Mul-LDA-DP algorithms can maintain good utility in SVM classification.

2. Related Work

In this section, we introduce the research status of privacy preserving data publishing in centralized scenario and distributed scenario, respectively.

2.1. Privacy Preserving Data Publishing in Centralized Scenario. Blum et al. [17] proposed the sublinear query (SULQ) input perturbation framework which adds noise to the covariance matrix, the framework can only be used for querying the projected subspace. Chaudhuri et al. [18] proposed the PPCA algorithm which is the improvement of SUQL algorithm. The PPCA algorithm randomly samples a k -dimensional subspace which ensures differential privacy and is biased toward high utility. Both SUQL and PPCA procedures are differentially private approximations to the top- k subspace. Zhang et al. [19] proposed the PrivBayes algorithm; first, they constructed a Bayesian network with

differential privacy, and then they used the Bayesian network to generate a data set for publication. Chen et al. [20] presented the JTree algorithm. First, they explored the relationship between the attributes based on the sparse vector sampling technology, and then they constructed a Markov network that satisfies differential privacy and generated a synthetic data set for publication. Zhang et al. [21] proposed the PrivHD algorithm based on the JTree. They used high-pass filtering techniques to speed up the construction of Markov network and built a better joint tree for generating synthetic data set for publication. Xu et al. [22] proposed the DPPro algorithm; first, they randomly projected the original high-dimensional data into a low-dimensional space, and then they added noise to the projection vector and low-dimensional projection data; finally, they released the low-dimensional projection data. Zhang et al. [23] presented the PrivMN method. They constructed a Markov model with differential privacy, and then used the Markov model to generate a synthetic data set for publication. The algorithms mentioned above are mainly used for privacy preserving data publishing in centralized scenarios.

2.2. Privacy Preserving Data Publishing in Distributed Scenario. There are fewer researches on privacy protection of horizontally partitioned data publication. Ge et al. [24] proposed a distributed principal component analysis (DPS-PCA) algorithm with differential privacy; first, data owners collaborated to analyze the principal components, while protecting the private information, and then they released low-dimensional subspaces of high-dimensional sparse data. Wang et al. [25] proposed an efficient and scalable protocol for computing principal components in a distributed environment. First, the data owner encrypted the shared data and sent them to the semitrusted third party, then the semitrusted third party performed a private aggregation algorithm on the encrypted data and sent the aggregated data to data user for calculating the principal components. Imtiaz et al. [26] presented a distributed principal component analysis (DPdisPCA) algorithm with differential privacy. Each data owner used Gaussian noise to perturbed the local covariance matrix, and with the assistance of a semitrusted third party to calculate the principal components while ensuring local data privacy. Alhadidi et al. [27] proposed a two-party data publishing algorithm with differential privacy. They first presented a two-party protocol for the exponential mechanism which can be used as a subprotocol, the data released by this algorithm are suitable for classification tasks. Cheng et al. [28] proposed a differential privacy sequential update of the Bayesian network algorithm which is called DP-SUBN³, data owners collaboratively constructed the Bayesian network, data owners can treat the intermediate results as prior knowledge to construct the Bayesian network, and then they used the Bayesian network to generate a data set for publication. Wang et al. [29] proposed a distributed differential privacy anonymous algorithm and guaranteed that each step of the algorithm satisfies the definition of secure two-party computation. This is the first research about differentially private data publishing for arbitrarily partitioned data. In our

prior work [16], we proposed the PPCA-DP-MH algorithm. First, data owners and a semitrusted third party cooperated to reduce the dimension of high-dimensional data to obtain the top k principal components that satisfy differential privacy, and then each data owner used the generative model of probabilistic principal component analysis to generate a data set with the same scale as the original data for publication. Different from the prior work [16], this paper uses the linear discriminant analysis to publish the projection data with differential privacy. Linear discriminant analysis can retain the class information of the data while reducing the dimension, which is beneficial to maintain the utility of the published data in classification.

3. Preliminaries

3.1. Linear Discriminant Analysis (LDA). Linear discriminant analysis proposed by Fisher is one of the most widely used and extremely effective methods in the field of dimensionality reduction and pattern recognition. Its typical applications include face recognition, target tracking and detection, credit card fraud detection, and speech recognition. The idea of linear discriminant analysis for binary classification is to choose the projection direction so that the samples of different classes after projection are as far apart as possible and the samples within each class are as clustered as possible. We denote the data set as $X = X^{(1)} \cup X^{(2)}$, $X^{(k)} = \{\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \dots, \mathbf{x}_{N^{(k)}}^{(k)}\}$, $k = 1, 2$. $N = N^{(1)} + N^{(2)}$. The within-class mean vector of samples in the original sample space is as follows:

$$\mu^{(k)} = \frac{1}{N^{(k)}} \sum_{\mathbf{x} \in X^{(k)}} \mathbf{x}, \quad k = 1, 2. \quad (1)$$

The between-class scatter matrix is as follows:

$$S_b = (\mu^{(1)} - \mu^{(2)})(\mu^{(1)} - \mu^{(2)})^T. \quad (2)$$

The within-class scatter matrix is as follows:

$$S^{(k)} = \sum_{\mathbf{x} \in X^{(k)}} \mathbf{x}\mathbf{x}^T - N^{(k)}\mu^{(k)}(\mu^{(k)})^T, \quad k = 1, 2. \quad (3)$$

Then, the pooled within-class scatter matrix is as follows:

$$S_w = S^{(1)} + S^{(2)}. \quad (4)$$

It can also be expressed as follows:

$$S_w = \sum_{k=1}^2 \sum_{\mathbf{x} \in X^{(k)}} \mathbf{x}\mathbf{x}^T - \sum_{k=1}^2 N^{(k)}\mu^{(k)}(\mu^{(k)})^T. \quad (5)$$

The criterion of Fisher is as follows:

$$\max_{\mathbf{w}} \frac{\mathbf{w}^T S_b^{-1} \mathbf{w}}{\mathbf{w}^T S_w^{-1} \mathbf{w}}. \quad (6)$$

Using the Lagrange multiplier method to find the optimal projection direction vector, we obtain the following:

$$\mathbf{w} = S_w^{-1}(\mu^{(1)} - \mu^{(2)}). \quad (7)$$

The result of linear discriminant analysis only gives the optimal projection direction, and does not give a clear classification result.

3.2. Differential Privacy. Differential privacy provides a rigorous privacy protection for sensitive information, it can be quantified by mathematical formulas. The essence of differential privacy is to use noise to randomly perturb the output results, so that it is difficult to distinguish the original input data according to the output results.

Definition 1. [30] A randomized algorithm \mathcal{M} is ε -indistinguishable if for any two neighboring databases D and \widehat{D} differing in a single entry, and for all $O \subseteq \text{Range}(\mathcal{M})$:

$$\left| \ln \frac{P_r[\mathcal{M}(D) \in O]}{P_r[\mathcal{M}(\widehat{D}) \in O]} \right| \leq \varepsilon, \quad (8)$$

where ε is a small positive real number.

When ε is small, $\ln(1 + \varepsilon) \approx \varepsilon$, so $P_r[\mathcal{M}(D) \in O]/P_r[\mathcal{M}(\widehat{D}) \in O] \in [1 - \varepsilon, 1 + \varepsilon]$, ε is used to control the probability ratio of algorithm \mathcal{M} to obtain the same output on two neighboring databases, which reflects the level of privacy protection that \mathcal{M} can provide.

Definition 2 [30]. A randomized algorithm \mathcal{M} is (ε, δ) differential privacy, if for any two neighboring databases D and \widehat{D} differing in a single entry, and for any $O(O \subseteq \text{Range}(\mathcal{M}))$ there is the following:

$$P_r\{\mathcal{M}(D) \in O\} \leq e^\varepsilon P_r\{\mathcal{M}(\widehat{D}) \in O\} + \delta, \quad (9)$$

where ε is a small positive real number called privacy budget and δ is a small positive real number. It is also called δ -approximate ε -indistinguishability.

Definition 3. is the relaxed version of differential privacy. When $\delta = 0$, it becomes Definition 1, which is the strict version of differential privacy. Formula (9) means that it is allowed to break the limit of formula (8) with a small probability δ .

Theorem 1 ([31]). *The sufficient condition for the random function \mathcal{M} to satisfy (ε, δ) differential privacy is as follows:*

$$P_r \left\{ \left| \ln \frac{P_r[\mathcal{M}(D) \in O]}{P_r[\mathcal{M}(\widehat{D}) \in O]} \right| > \varepsilon \right\} \leq \delta, \quad O \subseteq \text{Range}(\mathcal{M}). \quad (10)$$

Theorem 2 (Sequential Composition) [31]. *Let \mathcal{M}_i be an $(\varepsilon_i, \delta_i)$ differentially private algorithm, $i = 1, 2, \dots, n$, then for the same data set D , the combined algorithm $\mathcal{M}(\mathcal{M}_1(D), \mathcal{M}_2(D), \dots, \mathcal{M}_n(D))$ is $(\sum_{i=1}^n \varepsilon_i, \sum_{i=1}^n \delta_i)$ differential privacy.*

Theorem 3 (Parallel Composition) [31]. *Let \mathcal{M}_i be an $(\varepsilon_i, \delta_i)$ differentially private algorithm, $i = 1, 2, \dots, n$, D_1, D_2, \dots, D_n are disjoint data sets, the combined algorithm $\mathcal{M}(\mathcal{M}_1(D_1), \mathcal{M}_2(D_2), \dots, \mathcal{M}_n(D_n))$ is $\max_{1 \leq i \leq n} (\varepsilon_i, \delta_i)$ differential privacy.*

Theorem 4 (Post Processing) [31]. Let $\mathcal{M}: D \rightarrow R$ be a randomized algorithm that is (ϵ, δ) differential privacy, let $f: R \rightarrow R'$ be an arbitrary mapping, then $f \circ \mathcal{M}: D \rightarrow R'$ is (ϵ, δ) differential privacy.

4. Proposed Methods

In this section, we will propose two algorithms which are called LDA-DP and Mul-LDA-DP. The LDA-DP algorithm is used for privacy protection of data publishing in the centralized scenario, and the Mul-LDA-DP algorithm is used for privacy protection of data publishing in the distributed scenario. Without loss of generality, we assume that all individual data in this paper are normalized to p -dimensional unit vectors.

4.1. LDA-DP Algorithm. In this section, we propose the LDA-DP algorithm for centralized data publishing.

4.1.1. Problem Statement and Algorithm Proposed. The data set X contains two classes of data individuals denoted as $X = X^{(1)} \cup X^{(2)}$, where $X^{(k)} = \{\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \dots, \mathbf{x}_{N^{(k)}}^{(k)}\}$, $k = 1, 2$. Our goal is to protect the privacy information of the original data from being leaked while publishing the projection data of the original data.

In order to solve this problem, we propose the LDA-DP algorithm, which is mainly divided into two stages. First, we use the Gaussian mechanism of differential privacy to perturb the within-class mean vectors $\mu^{(k)}$ ($k = 1, 2$). Second, we use the Gaussian mechanism to perturb the pooled

within-class scatter matrix S_w . Finally, we get the projection direction vector \mathbf{w} that satisfies (ϵ, δ) differential privacy and publish the low-dimensional projected data of the original data. The specific details are in Algorithm 1.

4.1.2. Privacy Analysis of LDA-DP Algorithm

Theorem 5. The within-class mean vector $\mu^{(k)}$ ($k = 1, 2$) in Algorithm 1 satisfies (ϵ_1, δ_1) differential privacy when each entry of $\mathbf{g}^{(k)}$ ($k = 1, 2$) is sampled from $N(0, \sigma_1^2)$, where $\sigma_1 \geq p^{3/2} \sqrt{\ln 2 / \pi \delta_1^2} + \sqrt{p^3 \ln 2 / \pi \delta_1^2 + 2\epsilon_1 / \epsilon_1}$, $0 < \delta_1 < \sqrt{2/\pi}$.

Proof. We denote the two neighboring data sets are $X = X^{(1)} \cup X^{(2)}$ and $\hat{X} = \hat{X}^{(1)} \cup \hat{X}^{(2)}$, where only one individual is different, without losing general assumption. Suppose the different individuals are in $X^{(1)}$ and $\hat{X}^{(1)}$, we denote them as $\mathbf{x}_{N^{(1)}} \neq \hat{\mathbf{x}}_{N^{(1)}}$, they are p -dimensional unit vector. We denote $\mathbf{a} = \sum_{\mathbf{x} \in X^{(1)}} \mathbf{x}$ and $\hat{\mathbf{a}} = \sum_{\mathbf{x} \in \hat{X}^{(1)}} \mathbf{x}$, let $\mathbf{c} = \mathbf{a} + \mathbf{g}^{(1)}$ and $\hat{\mathbf{c}} = \hat{\mathbf{a}} + \hat{\mathbf{g}}^{(1)}$, each entry of $\mathbf{g}^{(1)}$ and $\hat{\mathbf{g}}^{(1)}$ is sampled from $N(0, \sigma_1^2)$.

The log ratio of the probabilities \mathbf{c} and $\hat{\mathbf{c}}$ at a point \mathbf{h} is $|\ln((P\{\mathbf{c} = \mathbf{h}|X\}) / (P\{\hat{\mathbf{c}} = \mathbf{h}|\hat{X}\}))|$, the numerator in the ratio describes the probability of seeing \mathbf{h} when the data set is X , the denominator corresponds the probability of seeing this same value when the data set is \hat{X} .

By Theorem 1, we will to find the value of σ_1 such that the inequality $|\ln((P\{\mathbf{c} = \mathbf{h}|X\}) / (P\{\hat{\mathbf{c}} = \mathbf{h}|\hat{X}\}))| = |\ln((P\{\mathbf{h} - \mathbf{a}|X\}) / (P\{\mathbf{h} - \hat{\mathbf{a}}|\hat{X}\}))| \leq \epsilon_1$ holds at least with probability $1 - \delta_1$.

$$\begin{aligned} \left| \ln \frac{P\{\mathbf{h} - \mathbf{a}|X\}}{P\{\mathbf{h} - \hat{\mathbf{a}}|\hat{X}\}} \right| &= \frac{1}{2\sigma_1^2} \left| \sum_{i=1}^p [(h_i - \hat{a}_i)^2 - (h_i - a_i)^2] \right| \\ &= \frac{1}{2\sigma_1^2} \sum_{i=1}^p \left| [2(h_i - a_i)(a_i - \hat{a}_i) + (a_i - \hat{a}_i)^2] \right| \\ &\leq \frac{1}{2\sigma_1^2} \sum_{i=1}^p 2|(h_i - a_i)(x_{N^{(1)i}} - \hat{x}_{N^{(1)i}})| + \frac{1}{2\sigma_1^2} \sum_{i=1}^p (x_{N^{(1)i}} - \hat{x}_{N^{(1)i}})^2. \end{aligned} \quad (11)$$

Using the Lagrange multiplier method, we can get the maximum value of the objective function $\sum_{i=1}^p (|x_{N^{(1)i}}| + |\hat{x}_{N^{(1)i}}|)$ is $2\sqrt{p}$ under the condition of $\sum_{i=1}^p (x_{N^{(1)i}})^2 = 1$, $\sum_{i=1}^p (\hat{x}_{N^{(1)i}})^2 = 1$.

Then, we can obtain: $\sum_{i=1}^p |(x_{N^{(1)i}}) - (\hat{x}_{N^{(1)i}})| \leq \sum_{i=1}^p (|x_{N^{(1)i}}| + |\hat{x}_{N^{(1)i}}|) \leq 2\sqrt{p}$. Similarly, we can obtain the following:

$$\sum_{i=1}^p (x_{N^{(1)i}}) - (\hat{x}_{N^{(1)i}})^2 \leq 4. \quad (12)$$

So, $1/2\sigma_1^2 \sum_{i=1}^p 2|(h_i - a_i)(x_{N^{(1)i}}) - (\hat{x}_{N^{(1)i}})| \leq 2p^{3/2}r/\sigma_1^2$, where $|g_i^{(1)}| = |h_i - a_i| \leq r$, for all i , and $1/2\sigma_1^2 \sum_{i=1}^p (x_{N^{(1)i}} - \hat{x}_{N^{(1)i}})^2 \leq 2/\sigma_1^2$.

Then, $|\ln((P\{\mathbf{b} = \mathbf{h}|X\}) / (P\{\hat{\mathbf{b}} = \mathbf{h}|\hat{X}\}))| \leq (2p^{3/2}r + 2)/\sigma_1^2$, this quantity is bounded by ϵ_1 whenever $|g_i^{(1)}| \leq r \leq (\epsilon_1\sigma_1^2 - 2)/2p^{3/2}$.

To ensure privacy loss bounded by ϵ_1 with probability at least $1 - \delta_1$, we require to find σ_1 that satisfies this inequality $P_r\{|g_i^{(1)}| \geq \epsilon_1\sigma_1^2 - 2/2p^{3/2}\} \leq \delta_1$, due to symmetry, we will find σ_1 such that $P_r\{g_i^{(1)} \geq (\epsilon_1\sigma_1^2 - 2)/2p^{3/2}\} \leq \delta_1/2$.

Input: Data sets X , privacy parameters $(\varepsilon_1, \delta_1)$, $(\varepsilon_2, \delta_2)$

Output: Projection direction vector \mathbf{w} , projection data \tilde{X}

- (1) **for** $k = 1$ to 2 **do**
- (2) Set $\sigma_1 = p^{3/2} \sqrt{\ln 2 / \pi \delta_1^2} + \sqrt{p^3 \ln 2 / \pi \delta_1^2 + 2\varepsilon_1 / \varepsilon_1}$, which generates a p dimension noise vector $\mathbf{g}^{(k)}$; each entry is sampled from $N(0, \sigma_1^2)$
- (3) Computes $\mu^{(k)} = 1/N^{(k)} (\sum_{\mathbf{x} \in X^{(k)}} \mathbf{x} + \mathbf{g}^{(k)})$
- (4) **end for**
- (5) **return** $\mu^{(k)}$, $k = 1, 2$
- (6) Set $\sigma_2 = (p+1) \sqrt{\ln 2 / \pi \delta_2^2} + \sqrt{(p+1)^2 \ln 2 / \pi \delta_2^2 + 4\varepsilon_2 / 2\varepsilon_2}$, which generates a $p \times p$ random matrix G . Let G be a symmetric matrix with the upper triangle (including the diagonal) entries are sampled from $N(0, \sigma_2^2)$ and make the symmetrical position entries in the lower triangle matrix equal to the upper triangle.
- (7) Computes $S_w = \sum_{\mathbf{x} \in X^{(k)}} \mathbf{x} \mathbf{x}^T - \sum_{k=1}^2 N^{(k)} \mu^{(k)} (\mu^{(k)})^T + G$
- (8) Computes $\mathbf{w} = S_w^{-1} (\mu^{(1)} - \mu^{(2)})$
- (9) Computes $\tilde{X} = X \mathbf{w}$

ALGORITHM 1: LDA-DP algorithm.

The tail bound is as follows:

$$\begin{aligned} P_r\{g_i^{(1)} > t\} &= \int_t^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{x^2}{2\sigma_1^2}} dx \stackrel{x=t+y}{=} \int_0^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(t+y)^2}{2\sigma_1^2}} dy, \\ &\leq \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{t^2}{2\sigma_1^2}} \int_0^{+\infty} e^{-\frac{ty}{\sigma_1^2}} dy \leq \frac{1}{t} \frac{\sigma_1}{\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma_1^2}}. \end{aligned} \quad (13)$$

We let $t = (\varepsilon_1 \sigma_1^2 - 2)/2p^{3/2}$, then $1/t\sigma_1 / \sqrt{2\pi} e^{-t^2/2\sigma_1^2} \leq \delta_1/2$, then we obtain the following:

$$\ln \frac{t}{\sigma_1} + \frac{t^2}{2\sigma_1^2} \geq \ln \frac{2}{\sqrt{2\pi}\delta_1}. \quad (14)$$

When $\sigma_1 \geq [2p^{3/2} + (\sqrt{4p^3 + 8\varepsilon_1})]/2\varepsilon_1$, the first term in (14) is non-negative. To make the inequality (14) hold, we let $t^2/2\sigma_1^2 = 1/2\sigma_1^2 (\varepsilon_1 \sigma_1^2 - 2/2p^{3/2})^2 \geq \ln 2 / \sqrt{2\pi}\delta_1$, then we obtain the following:

$$\sigma_1 \geq \frac{p^{(3/2)} \sqrt{\ln(2/\pi\delta_1^2)} + \sqrt{p^3 \ln(2/\pi\delta_1^2) + 2\varepsilon_1}}{\varepsilon_1}, \quad 0 < \delta_1 < \sqrt{\frac{2}{\pi}}. \quad (15)$$

Theorem 6. *The pooled within-class scatter matrix S_w in Algorithm 1 satisfies $(\varepsilon_2, \delta_2)$ differential privacy, when each entry in the symmetric random matrix G is sampled from $N(0, \sigma_2^2)$, where*

$$\begin{aligned} \sigma_2 &\geq (p+1) \sqrt{\ln \frac{2}{\pi\delta_2^2}} + \sqrt{(p+1)^2 \ln \frac{2}{\pi\delta_2^2} + 4\varepsilon_2/2\varepsilon_2}, \\ 0 < \delta_2 &< \sqrt{\frac{2}{\pi}}. \end{aligned} \quad (16)$$

Proof. Two neighboring data sets are $X = X^{(1)} \cup X^{(2)}$ and $\hat{X} = \hat{X}^{(1)} \cup \hat{X}^{(2)}$, where only one entry is different, without losing general assumption, suppose the different entry are in $X^{(1)}$ and $\hat{X}^{(1)}$ and denoted them as $\mathbf{x}_{N^{(1)}} \neq \hat{\mathbf{x}}_{N^{(1)}}$.

Because $\mu^{(k)}$ ($k = 1, 2$) in (5) satisfies differential privacy has been proved by Theorem 5 which can be treated as a constant in (5), so if we want to prove that this theorem holds, it is only necessary to prove that the first item $\sum_{k=1}^2 \sum_{\mathbf{x} \in X^{(k)}} \mathbf{x} \mathbf{x}^T$ in (5) satisfies $(\varepsilon_2, \delta_2)$ differential privacy after adding random matrix G .

We denote $B = \sum_{k=1}^2 \sum_{\mathbf{x} \in X^{(k)}} \mathbf{x} \mathbf{x}^T$, $\hat{B} = \sum_{k=1}^2 \sum_{\mathbf{x} \in \hat{X}^{(k)}} \mathbf{x} \mathbf{x}^T$, let $C = B + G$ and $\hat{C} = \hat{B} + G$, G and \hat{G} are two independent symmetric random matrices with the upper triangle (including the diagonal) entries are sampled from $N(0, \sigma_2^2)$, and make the symmetrical position entries in the lower triangle matrix equal to the upper triangle.

The log ratio of the probabilities C and \hat{C} at a point H is $|\ln((P\{C = H|X\})/(P\{\hat{C} = H|\hat{X}\}))|$.

By Theorem 1, we need to find the value of σ_2 such that the inequality $|\ln((P\{C = H|X\})/(P\{\hat{C} = H|\hat{X}\}))| \leq \varepsilon_2$ holds at least with probability $1 - \delta_2$.

$$\begin{aligned} \left| \ln \frac{P\{C = H|X\}}{P\{\hat{C} = H|\hat{X}\}} \right| &= \left| \ln \frac{P\{H - B|X\}}{P\{H - \hat{B}|\hat{X}\}} \right| \\ &= \frac{1}{2\sigma_2^2} \left| \sum_{1 \leq i \leq j \leq p} [(H_{ij} - \hat{B}_{ij})^2 - (H_{ij} - B_{ij})^2] \right| \\ &= \frac{1}{2\sigma_2^2} \left| \sum_{1 \leq i \leq j \leq p} [2(H_{ij} - B_{ij})(B_{ij} - \hat{B}_{ij}) + (B_{ij} - \hat{B}_{ij})^2] \right| \end{aligned}$$

$$\begin{aligned} &\leq \frac{1}{2\sigma_2^2} \sum_{1 \leq i \leq j \leq p} 2 \left| (H_{ij} - B_{ij})(x_{N^{(i)}} x_{N^{(j)}} - \hat{x}_{N^{(i)}} \hat{x}_{N^{(j)}}) \right| \\ &\quad + \frac{1}{2\sigma_2^2} \sum_{1 \leq i \leq j \leq p} (x_{N^{(i)}} x_{N^{(j)}} - \hat{x}_{N^{(i)}} \hat{x}_{N^{(j)}})^2. \end{aligned} \quad (17)$$

By using the Lagrange multiplier method and the inequality in [18], the following inequalities hold:

$$\begin{aligned} &\sum_{1 \leq i \leq j \leq p} (x_{N^{(i)}} x_{N^{(j)}} - \hat{x}_{N^{(i)}} \hat{x}_{N^{(j)}})^2 \leq 2, \\ &\sum_{1 \leq i \leq j \leq p} \left| (x_{N^{(i)}} x_{N^{(j)}} - \hat{x}_{N^{(i)}} \hat{x}_{N^{(j)}}) \right| \leq p + 1. \end{aligned} \quad (18)$$

Then, $|\ln((P\{H - B|X\})/(P\{H - \hat{B}|\hat{X}\}))| \leq r(p + 1) + 1/\sigma_2^2$, where $|G_{ij}| = |H_{ij} - B_{ij}| \leq r$ for all i, j .

The rest of the proof process is similar to Theorem 5, then we can obtain the following:

$$\sigma_2 \geq \frac{(p + 1) \sqrt{\ln(2/\pi\delta_2^2)} + \sqrt{(p + 1)^2 \ln(2/\pi\delta_2^2) + 4\epsilon_2}}{2\epsilon_2}, \quad (19)$$

$$0 < \delta_2 < \sqrt{\frac{2}{\pi}}.$$

We have proven that the within-class mean vector $\mu^{(k)}$ ($k = 1, 2$) satisfies (ϵ_1, δ_1) differential privacy, the pooled within-class scatter matrix S_w satisfies (ϵ_2, δ_2) differential privacy, by the property of differential privacy sequential composition, the projection direction vector in the Algorithm 1 satisfies (ϵ, δ) differential privacy, where $\epsilon = \epsilon_1 + \epsilon_2, \delta = \delta_1 + \delta_2$. For the published projection data $\tilde{X} = X\mathbf{w}$, $X \in R^{N \times p}$, $\mathbf{w} \in R^{p \times 1}, p < N$, we can regard $\tilde{X} = X\mathbf{w}$ as a set of undetermined system of equation, the number of variables are more than equations, so the equation has infinitely many sets of solutions, that is, it is impossible to infer the information of the original data X from the published projection data \tilde{X} .

4.2. Mul-LDA-DP Algorithm. In this section, we propose the Mul-LDA-DP algorithm for distributed data publishing. The mathematical notations used in this section are summarized in Table 2.

4.2.1. Problem Statement and Algorithm Proposed. In the distributed scenario, data are stored by multiple data owners rather than a single owner, and the data owners do not trust each other. Data at a single site may not be sufficient for statistical learning. One solution is that each data owner uses the LDA-DP algorithm in Section 4.1 to publish the projection data independently. Another solution is the data owners cooperate with each other to publish the projection data of the integrated data. Comparing the two solutions, it is obvious that the latter solution can improve the utility of publishing data. Based on the idea of the second solution and [32], we propose the Mul-LDA-DP algorithm for distributed data publishing. The entity description of the model is as follows.

- (1) Data owner. The data owner P_m ($m = 1, 2, \dots, M$) has a data set X_m . Each data owner can generate random vectors and matrices to perturb the within-class mean vectors and within-class scatter matrices locally.
- (2) Data publisher. The data publisher is a data publishing platform based on blockchain. The data publisher aggregates the local within-class mean vectors and within-class scatter matrices with noise. The data publisher can obtain the projection vector that satisfies differential privacy and publishes the projection data of the pooled data.
- (3) A random number generator. It can generate random vectors and random matrices and send them to data owners and data publisher secretly.

Threat Model. In our setting, we assume that the data owners and data publisher are honest-but-curious, that is, they follow the protocol but may try to deduce information of other data owners from the received messages.

Two types of adversaries are considered, which are external attackers and internal attackers. External attackers which can be called an external eavesdropper may gain access to information such as data sent by data owners to the data publisher. Internal adversaries can be the data owners and the data publisher. The goal of each data owner is to extract the information not owned by him, while the goal of the data publisher is to extract the information from each data owner.

Distributed Within-Class Mean Vectors and Pooled Within-Class Scatter Matrix Computation. When the data are owned by M data owners, the within-class mean vectors (1) can be decomposed into the following:

$$\mu^{(k)} = \frac{1}{N^{(k)}} \sum_{m=1}^M N_m^{(k)} \mu_m^{(k)}, \quad k = 1, 2, \quad (20)$$

$$\text{where } \mu_m^{(k)} = 1/N_m^{(k)} \sum_{\mathbf{x} \in X_m^{(k)}} \mathbf{x}.$$

The pooled within-class scatter matrix (5) can be decomposed into the following:

$$S_w = \sum_{k=1}^2 S^{(k)} = \sum_{k=1}^2 \sum_{m=1}^M S_m^{(k)} = \sum_{m=1}^M \sum_{k=1}^2 S_m^{(k)}, \quad (21)$$

$$\text{where } S_m^{(k)} = \sum_{\mathbf{x} \in X_m^{(k)}} \mathbf{x}\mathbf{x}^T - N_m^{(k)} \mu_m^{(k)} (\mu_m^{(k)})^T.$$

The abovementioned result allows each data owner to compute and perturb a partial result simultaneously locally. Therefore, we use the additivity of Gaussian distribution to

TABLE 2: Summary of notations.

Notation	Explanation
M	The number of data owners
P_m	The m -th data owner
$N_m^{(k)}$	The number of individuals in the k -th class owned by P_m
$N^{(k)}$	The total number of individuals in the k -th class, $N^{(k)} = \sum_{m=1}^M N_m^{(k)}$
$X_m^{(k)}$	The set of the k -th class data owned by P_m
$X^{(k)}$	The set of the k -th class data. $X^{(k)} = \cup_{m=1}^M X_m^{(k)}$
X_m	The data set owned by P_m . $X_m = \cup_{k=1}^2 X_m^{(k)}$
$\mu_m^{(k)}$	The within-class mean vector of the k -th class data owned by P_m
$\mu^{(k)}$	The within-class mean vector of the k -th class data
$S_m^{(k)}$	The within-class scatter matrix of the k -th class data owned by P_m
$S^{(k)}$	The within-class scatter matrix of the k -th class data
S_w	The pooled within-class scatter matrix

propose a correlated noise generation scheme. We design the noise generation procedure such that (i) we can ensure that the data output from each data owner satisfy differential privacy and (ii) we can achieve the noise level of the same as the pooled data scenario.

Scheme for Perturbing Shared Data by Correlated Noise. To prevent the data publisher and other data owners learning the privacy of local data, the data owner uses the noise generated by himself and the noise generated by the random number generator to perturb the local within-class mean vectors and within-class scatter matrices. Through our correlated noise design scheme, the data aggregated by the data publisher contain the same level of noise as the centralized scenario. The scheme is described as below:

- (1) Initialization stage. The random number generator generates p dimensional random vectors $\bar{\mathbf{g}}_m^{(k)}$, each entry is sampled from $N(0, (M-1)/M\sigma_1^2)$, generates $p \times p$ random matrices \bar{G}_m , let \bar{G}_m be the symmetric matrix with the upper triangle (including the diagonal) entries are sampled from $N(0, (M-1)/M\sigma_2^2)$, and makes the symmetrical position entries in the lower triangle matrix equal to the upper triangle, $m = 0, 1, 2, \dots, M, k = 1, 2$. Make these random vectors and matrices satisfy $\sum_{m=0}^M \bar{\mathbf{g}}_m^{(k)} = 0$, $\sum_{m=0}^M \bar{G}_m = (0)_{p \times p}$, then $\bar{\mathbf{g}}_m^{(k)}$ ($k = 1, 2$) and \bar{G}_m are sent to data owner P_m secretly, $\bar{\mathbf{g}}_0^{(k)}$ ($k = 1, 2$) and \bar{G}_0 are sent to the data publisher secretly.
- (2) Data owner P_m generates p dimensional random vectors $\mathbf{g}_m^{(k)}$ ($k = 1, 2$), each entry is sampled from $N(0, 1/M\sigma_1^2)$, computes $\mu_m^{(k)} = 1/N_m^{(k)} (\sum_{\mathbf{x} \in X_m^{(k)}} \mathbf{x} + \mathbf{g}_m^{(k)} + \bar{\mathbf{g}}_m^{(k)})$, $k = 1, 2$, and sends them to the data publisher.
- (3) The data publisher computes $\mu^{(k)} = 1/N^{(k)} (\sum_{m=1}^M N_m^{(k)} \mu_m^{(k)} + \bar{\mathbf{g}}_0^{(k)})$, $k = 1, 2$ and sends them to each data owner.
- (4) The data owner P_m generates $p \times p$ random matrix G_m , let G_m be the symmetric matrix with the upper triangle (including the diagonal) entries are sampled from $N(0, 1/M\sigma_2^2)$, and make the symmetrical position entries in the lower triangle matrix equal to the

upper triangle. Data owner P_m computes $S_m = \sum_{k=1}^2 S_m^{(k)} + G_m + \bar{G}_m$ and sends it to the data publisher.

- (5) The data publisher computes $S_w = \sum_{m=1}^M S_m + \bar{G}_0$ and calculates the projection vector \mathbf{w} that satisfies differential privacy.

The specific details of Mul-LDA-DP algorithm are in Algorithm 2. The input random vectors $\bar{\mathbf{g}}_m^{(k)}$ and random matrices \bar{G}_m in Algorithm 2 are generated in the initialization stage by the random number generator, $m = 0, 1, 2, \dots, M, k = 1, 2$.

4.2.2. Privacy Analysis of the Mul-LDA-DP Algorithm

Theorem 7. *The within-class mean vector $\mu^{(k)}$ ($k = 1, 2$) in Algorithm 2 satisfies (ϵ_1, δ_1) differential privacy.*

Proof. $\mu_m^{(k)} = 1/N_m^{(k)} (\sum_{\mathbf{x} \in X_m^{(k)}} \mathbf{x} + \mathbf{g}_m^{(k)} + \bar{\mathbf{g}}_m^{(k)})$ because each entry of $\mathbf{g}_m^{(k)}$ is sampled from $N(0, 1/M\sigma_1^2)$, and each entry of $\bar{\mathbf{g}}_m^{(k)}$ is sampled from $N(0, M-1/M\sigma_1^2)$, so each entry of $\mathbf{g}_m^{(k)} + \bar{\mathbf{g}}_m^{(k)}$ obeys $N(0, \sigma_1^2)$. By Theorem 5, $\mu_m^{(k)}$ satisfies (ϵ_1, δ_1) differential privacy.

Due to the post-processing property of differential privacy, the within-class mean vector $\mu^{(k)} = 1/N^{(k)} (\sum_{m=1}^M N_m^{(k)} \mu_m^{(k)} + \bar{\mathbf{g}}_0^{(k)})$ in Algorithm 2 satisfies (ϵ_1, δ_1) differential privacy.

Theorem 8. *The pooled within-class scatter matrix S_w in Algorithm 2 satisfies (ϵ_2, δ_2) differential privacy.*

Proof. $S_m = \sum_{k=1}^2 S_m^{(k)} + G_m + \bar{G}_m$, where each entry of symmetric random matrix G_m is sampled from $N(0, 1/M\sigma_2^2)$, and each entry of symmetric random matrix \bar{G}_m is sampled from $N(0, (M-1)/M\sigma_2^2)$, so each entry of $G_m + \bar{G}_m$ obeys $N(0, \sigma_2^2)$. By Theorem 6, S_m satisfies (ϵ_2, δ_2) differential privacy. Due to the post-processing property of differential privacy, the pooled within-class scatter matrix $S_w = \sum_{m=1}^M S_m + \bar{G}_0$ in Algorithm 2 satisfies (ϵ_2, δ_2) differential privacy.

We have proven both $\mu^{(k)}$ ($k = 1, 2$) and S_w satisfy differential privacy, we will show that the level of noise is the same as the centralized scenario. In the initialization stage,

Input: Data sets $X_m, m = 1, 2, \dots, M, k = 1, 2$, privacy parameters $(\epsilon_1, \delta_1), (\epsilon_2, \delta_2)$, random vector $\bar{\mathbf{g}}_m^{(k)}$ and random matrix \bar{G}_m which are generated in initialization stage, $m = 0, 1, 2, \dots, M; k = 1, 2$.

Output: Projection direction vector \mathbf{w} , projection data \tilde{X}

- (1) **for** $m = 1$ to M **do**
- (2) **for** $k = 1$ to 2 **do**
- (3) Set $\sigma_1 = p^{3/2} \sqrt{\ln 2 / \pi \delta_1^2} + \sqrt{p^3 \ln 2 / \pi \delta_1^2 + 2\epsilon_1 / \epsilon_1}$, data owner generates p dimensional random vector $\mathbf{g}_m^{(k)}$, each entry is sampled from $N(0, \sigma_1^2 / M)$
- (4) Compute $\mu_m^{(k)} = 1 / N_m^{(k)} (\sum_{\mathbf{x} \in X_m^{(k)}} \mathbf{x} + \mathbf{g}_m^{(k)} + \bar{\mathbf{g}}_m^{(k)})$
- (5) **end for**
- (6) **end for**
- (7) Compute $\mu^{(k)} = 1 / N^{(k)} (\sum_{m=1}^M N_m^{(k)} \mu_m^{(k)} + \bar{\mathbf{g}}_0^{(k)})$
- (8) **for** $m = 1$ to M **do**
- (9) Set $\sigma_2 = (p + 1) \sqrt{\ln 2 / \pi \delta_2^2} + \sqrt{(p + 1)^2 \ln 2 / \pi \delta_2^2 + 4\epsilon_2 / 2\epsilon_2}$, data owner generates $p \times p$ symmetric random matrices G_m , each entry is sampled from $N(0, \sigma_2^2 / M)$
- (10) **for** $k = 1$ to 2 **do**
- (11) Compute $S_m^{(k)} = \sum_{\mathbf{x} \in X_m^{(k)}} \mathbf{x} \mathbf{x}^T - N_m^{(k)} \mu_m^{(k)} (\mu_m^{(k)})^T$
- (12) **end for**
- (13) Compute $S_m = \sum_{k=1}^2 S_m^{(k)} + G_m + \bar{G}_m$
- (14) **end for**
- (15) Compute $S_w = \sum_{m=1}^M S_m + \bar{G}_0$
- (16) Compute $\mathbf{w} = S_w^{-1} (\mu^{(1)} - \mu^{(2)})$
- (17) **return** $\tilde{X} = \cup_{m=1}^M X_m \mathbf{w}$

ALGORITHM 2: Mul-LDA-DP algorithm.

the noise vectors and matrices generated by the random number generator satisfy $\sum_{m=0}^M \bar{\mathbf{g}}_m^{(k)} = \mathbf{0}$ and $\sum_{m=0}^M \bar{G}_m = (\mathbf{0})_{p \times p}$.

The within-class mean vector $\mu^{(k)}$ ($k = 1, 2$) is as follows:

$$\begin{aligned}
 \mu^{(k)} &= \frac{1}{N^{(k)}} \left(\sum_{m=1}^M N_m^{(k)} \mu_m^{(k)} + \bar{\mathbf{g}}_0^{(k)} \right) = \frac{1}{N^{(k)}} \left[\sum_{m=1}^M \left(\sum_{\mathbf{x} \in X_m^{(k)}} \mathbf{x} + \mathbf{g}_m^{(k)} + \bar{\mathbf{g}}_m^{(k)} \right) + \bar{\mathbf{g}}_0^{(k)} \right], \\
 &= \frac{1}{N^{(k)}} \left[\sum_{m=1}^M \sum_{\mathbf{x} \in X_m^{(k)}} \mathbf{x} + \sum_{m=1}^M \mathbf{g}_m^{(k)} + \sum_{m=0}^M \bar{\mathbf{g}}_m^{(k)} \right] = \frac{1}{N^{(k)}} \left[\sum_{m=1}^M \sum_{\mathbf{x} \in X_m^{(k)}} \mathbf{x} + \sum_{m=1}^M \mathbf{g}_m^{(k)} \right], \\
 &= \frac{1}{N^{(k)}} \sum_{\mathbf{x} \in X^{(k)}} \mathbf{x} + \sum_{m=1}^M \mathbf{g}_m^{(k)}.
 \end{aligned} \tag{22}$$

Each entry of $\sum_{m=1}^M \mathbf{g}_m^{(k)}$ obeys $N(0, \sigma_1^2)$.

The pooled within-class scatter matrix S_w is as follows:

$$\begin{aligned}
 S_w &= \sum_{m=1}^M S_m + \bar{G}_0 = \sum_{m=1}^M \left(\sum_{k=1}^2 S_m^{(k)} + G_m + \bar{G}_m \right) + \bar{G}_0, \\
 &= \sum_{m=1}^M \sum_{k=1}^2 S_m^{(k)} + \sum_{m=1}^M G_m + \sum_{m=0}^M \bar{G}_m = \sum_{k=1}^2 \sum_{m=1}^M S_m^{(k)} + \sum_{m=1}^M G_m, \\
 &= \sum_{k=1}^2 S^{(k)} + \sum_{m=1}^M G_m.
 \end{aligned} \tag{23}$$

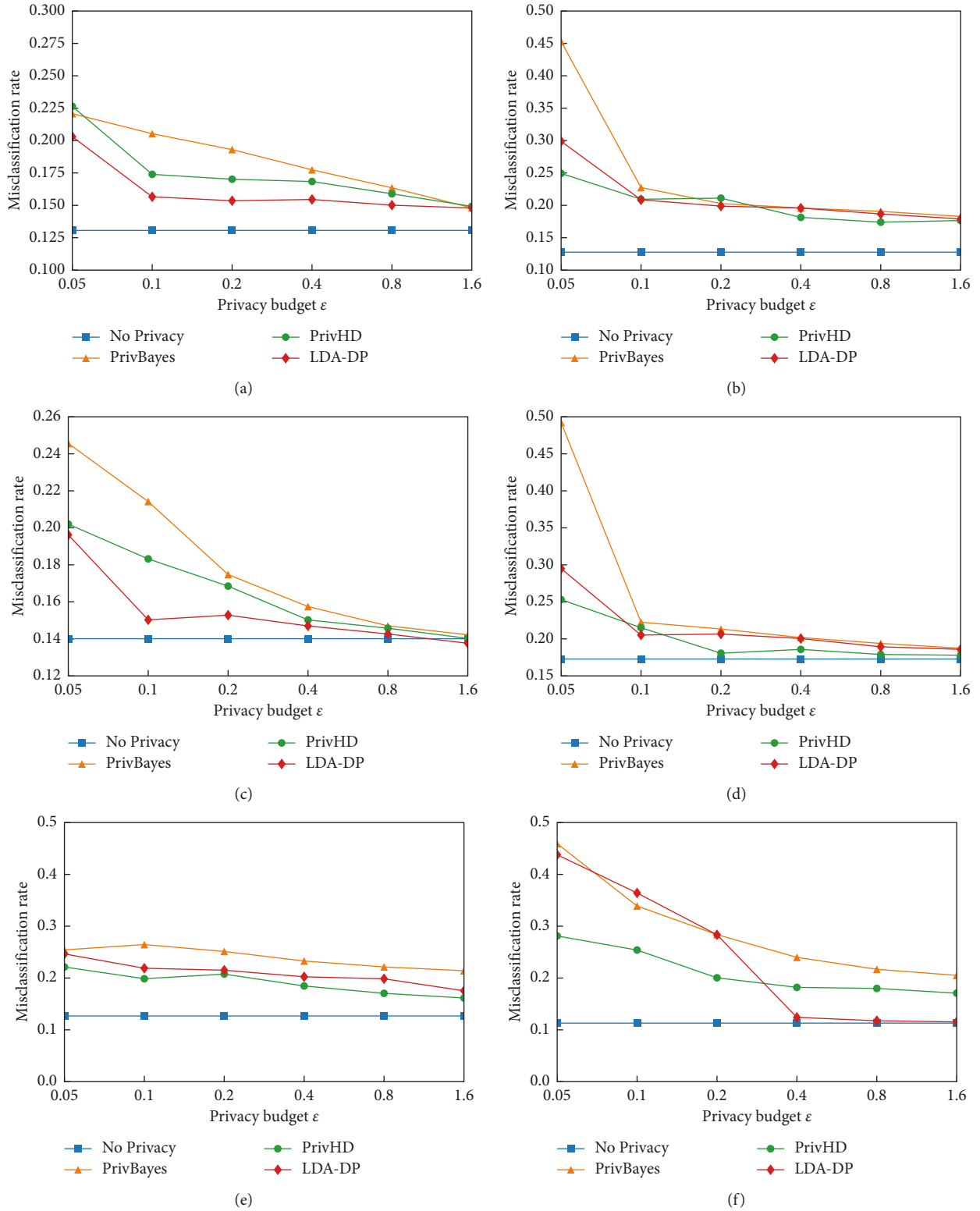


FIGURE 1: SVM misclassification rate of LDA-DP, PrivBayes and PRivHD under different privacy budgets. (a) NLTCS, Y =money. (b) NLTCS, Y =outside. (c) NLTCS, Y =bathing. (d) NLTCS, Y =travelling. (e) Adult, Y =salary. (f) Adult, Y =education.

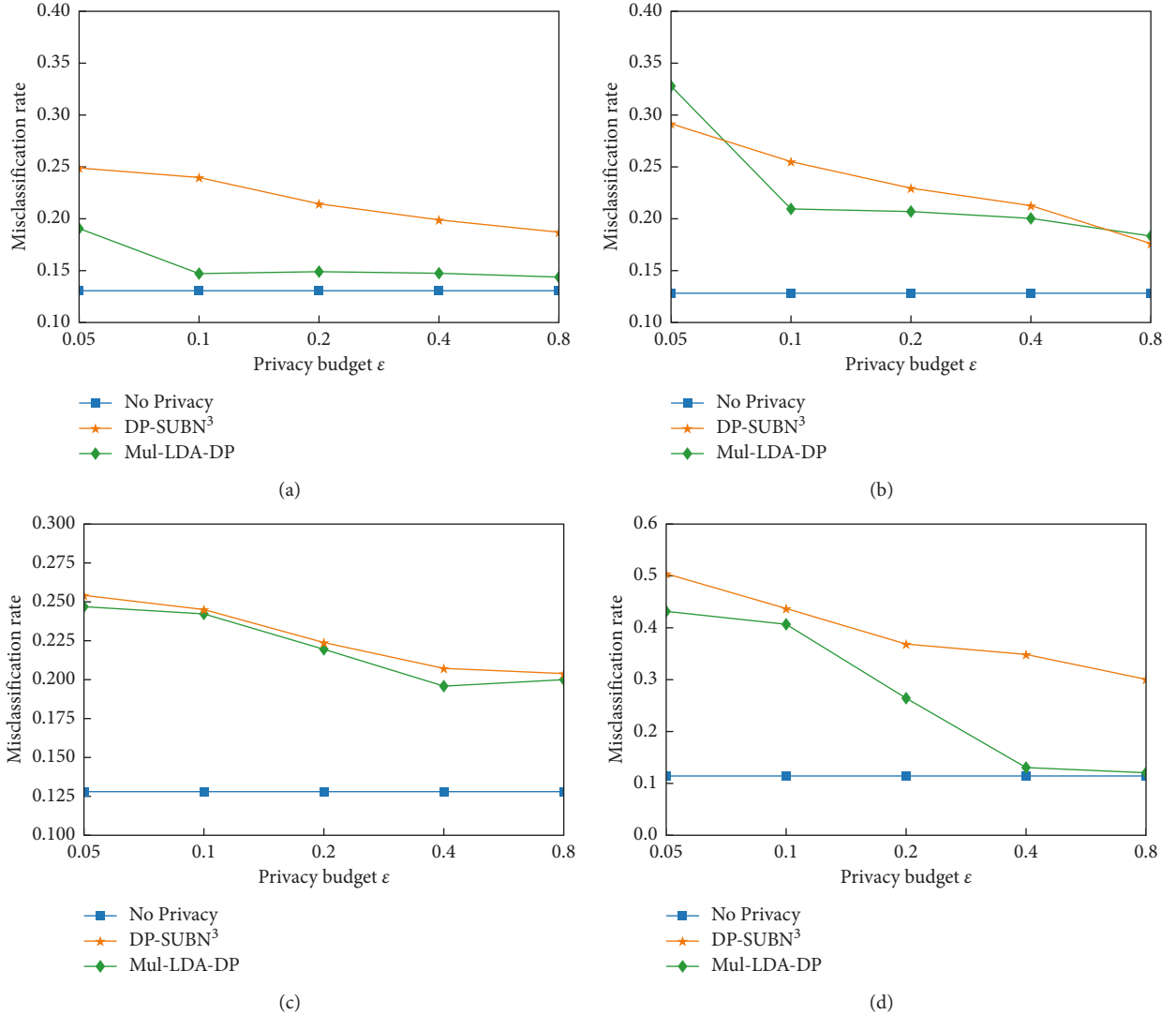


FIGURE 2: SVM misclassification rate of Mul-LDA-DA and DP-SUBN³ under different privacy budgets. (a) NLTCS, $Y = \text{money}$. (b) NLTCS, $Y = \text{outside}$ (c) Adult, $Y = \text{salary}$. (d) Adult, $Y = \text{education}$.

Each entry of $\sum_{m=1}^M G_m$ obeys $N(0, \sigma_2^2)$.

According to Theorems 5 and 6, the within-class mean vector $\mu^{(k)}$ ($k = 1, 2$) and pooled within-class scatter matrix S_w contain the same level of noise as the centralized scenario, and we achieve the purpose of improving the utility of publishing data while protecting the data privacy.

There are three opportunities for attackers to steal the data transmitted between the data owner and the data publisher. The first time is that the data owner sends the within-class mean vectors to the data publisher, the second time is that the data owner sends the within-class scatter matrices to data publisher. From Theorems 7 and 8, we know that the within-class mean vectors and the within-class scatter matrices satisfy differential privacy. Therefore, the attacker cannot infer the information of the original data from the eavesdropped data. The third time is that the data owner sends projection data to the data publisher, in Section 4.1.2, we have analyzed that it is impossible to infer the information of the original data from the published projection data.

5. Experiment

In order to measure the usability of the LDA-DP and Mul-LDA-DP algorithms proposed in this paper, we conduct experiments on real data sets which are Adult and NLTCS. Adult data set is extracted from the 1994 US Census, it contains 45222 individuals, each individual has 15 attributes. NLTCS data set is extracted from the National Long Term Care Survey, and recorded the daily activities of 21574 disabled persons at different time periods, each individual has 16 attributes. We use the SVM misclassification rate to measure the availability of the published data. For the Adult data set, it is necessary to predict whether a person (1) holds a post-secondary degree and (2) earns more than 50K. For the NLTCS data set, we need to predict whether a person (1) is unable to get outside, (2) is unable to manage money, (3) is unable to travel, and (4) is unable to bath. In our experiments, we set $\delta = 0.001$ to remain unchanged, and ϵ to take different values. We uniformly divide the privacy parameters

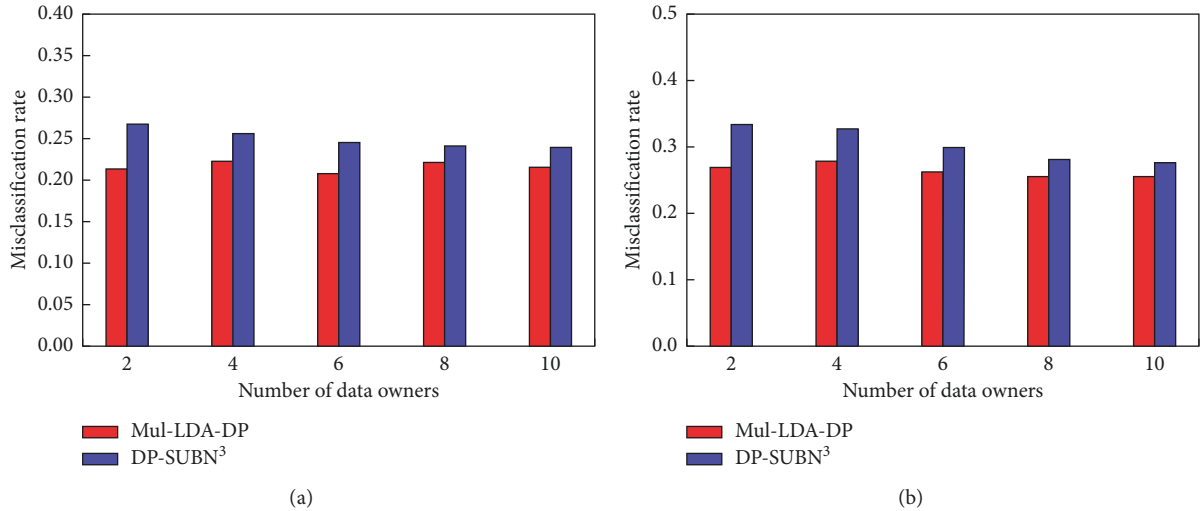


FIGURE 3: SVM misclassification rate of Mul-LDA-D and DP-SUBN³ under different number of data owners. (a) Adult, $Y = \text{salary}$. (b) Adult, $Y = \text{education}$.

into 2 portions ($\varepsilon_1 = \varepsilon_2 = \varepsilon/2, \delta_1 = \delta_2 = \delta/2$). Each experiment was repeated 50 times, and the mean value was taken as the experimental result. We use “No Privacy” to represent the SVM misclassification rate on the original data set.

5.1. Comparing the Performance of LDA-DA, PrivBayes, and PRivHD Algorithms under Different Privacy Budgets. The LDA-DA, PrivBayes, and PrivHD algorithms are all suitable for the centralized data publishing scenario, so in this set of experiments, we set the number of data owners to 1, and privacy budget ε takes different values. As can be seen from Figure 1, for both Adult and NLTCS data sets, the SVM classification utility of the data published by the LDA-DP algorithm outperforms the PrivBayes algorithm. The LDA-DP algorithm outperforms the PrivHD algorithm on the NLTCS dataset; however, the LDA-DP algorithm has slightly lower SVM classification utility on the Adult dataset than the PrivHD algorithm. We can also observe a commonality, for LDA-DA, PrivBayes, and PRivHD algorithms, the SVM misclassification rate decreases with the increase of the privacy budget ε . This phenomenon is consistent with the theory that as the privacy budget ε increases, privacy protection will weaken and the availability of data will increase.

5.2. Comparing the Performance of Mul-LDA-DA and DP-SUBN³ Algorithms under Different Privacy Budgets. The algorithm Mul-LDA-DP proposed in this paper is suitable for the distributed data publishing scenario, so in this set of experiments, we set the number of data owners to 3, and privacy budget ε takes different values. We train classifiers on published data set to compare the efficacy of Mul-LDA-DA and DP-SUBN³ algorithms. From Figure 2, we can see that the SVM classification utility of the data published by the Mul-LDA-DP algorithm outperforms the DP-SUBN³ algorithm. Both on money of NLTCS and education of Adult classifiers, the misclassification rate of Mul-LDA-DA

algorithm is significantly lower than the DP-SUBN³ algorithm especially.

5.3. Comparing the Performance of Mul-LDA-DA and DP-SUBN³ Algorithms under Different Number of Data Owners. In this section, the experiment studied the relationship between SVM misclassification rate and the number of data owners. The number of data owners is set to 2, 4, 6, 8, 10, and the privacy budget ε is set to 0.2, We trained two classifiers, education classifier, and salary classifier on Adult data set. The results in Figure 3 show that the SVM misclassification rate of the Mul-LDA-DP algorithm remains stable with the change of the number of data owners. The reason is that we perturb the local shared data by generating correlated noise based on the additivity of the Gaussian distribution. This scheme ensures that the level of Gaussian noise added to the data in the distributed scenario is similar to the noise level in the centralized scenario. Therefore, as the number of data owners increases, the misclassification rate remains stable. The SVM misclassification rate of DP-SUBN³ algorithm decreases as the number of data owners increases. This is because as the number of data owners increases, the number of update iterations increases when constructing the Bayesian network, and the Bayesian network constructed is closer to the distribution of the original data. However, from Figure 3, we can see that the performance of Mul-LDA-DA algorithm is still better than DP-SUBN³ algorithm when the number of data owners is no more than 10.

6. Conclusion

In this paper, we propose two algorithms for privacy preserving data publishing, the LDA-DP algorithm for data publishing in the scenario, and the Mul-LDA-DP algorithm for multiparty horizontally split data publishing. We use the additivity of Gaussian distribution to alleviate the effects of noise and can achieve the same noise level as the centralized

scenario. The experimental results show that the projection data released by the two algorithms can maintain high utility in SVM classification. However, the research in this paper also has limitations. 1) We only research the privacy protection problem when the data are a binary classification, but they are often multiclassification data. 2) The data released by the two algorithms in this paper are low-dimensional projection data of the original data, which limit the analysis and mining of the released data in many aspects. In the future, we will continue to conduct research on the abovementioned issues.

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.

References

- [1] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "Psspr: a source location privacy protection scheme based on sector phantom routing in wsns," *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1204–1221, 2021.
- [2] Q. Liu, J. Yu, J. Han, and X. Yao, "Differentially private and utility-aware publication of trajectory data," *Expert Systems with Applications*, vol. 180, no. 7, Article ID 115120, 2021.
- [3] L. Sweeney and L. K. Anonymity, "A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, 2002.
- [4] R. Rongxing Lu, X. Xiaohui Liang, L. Xu Li, X. Xiaodong Lin, and X. Xuemin Shen, "Eppa: an efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.
- [5] C. Wang, D. Wang, G. Xu, and D. He, "Efficient Privacy-Preserving User Authentication Scheme with Forward Secrecy for Industry 4.0," *SCIENCE CHINA: Information Sciences*, vol. 65, Article ID 112301, 2020.
- [6] C.-T. Li, D.-H. Shih, C.-C. Wang, C.-L. Chen, and C.-C. Lee, "A blockchain based data aggregation and group authentication scheme for electronic medical system," *IEEE Access*, vol. 8, Article ID 173904, 2020.
- [7] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, vol. 36, 2021.
- [8] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
- [9] Y.-T. Tsou and B.-C. Lin, "PPDCA: privacy-preserving crowdsourcing data collection and analysis with randomized response," *IEEE Access*, vol. 6, Article ID 76970, 2018.
- [10] X. Ren, C.-M. Yu, W. Yu et al., "High-dimensional Crowd-sourced Data Publication with Local Differential Privacy," *IEEE Transactions on Information Forensics & Security*, vol. 13, no. 9, pp. 2151–2166, 2018.
- [11] Y. Chen, D. Sen, T. Li, Y. Wang, and H. Zhou, "Dynamic multi-key fhe in asymmetric key setting from lwe," *IEEE Transactions on Information Forensics and Security*, vol. 16, no. 1–1, 2021.
- [12] Q. Wang, Y. Zhang, L. Xiao, Z. Wang, and K. Ren, "Rescuedp: real-time spatio-temporal crowd-sourced data publishing with differential privacy," in *Proceedings of the IEEE Infocom - the IEEE International Conference on Computer Communications*, San Francisco, CA, USA, April, 2016.
- [13] W. Hao and Z. Xu, "Cts-dp: publishing correlated time-series data via differential privacy," *Knowledge-Based Systems*, vol. 122, pp. 167–179, 2017.
- [14] H. Wang and H. Wang, "Correlated tuple data release via differential privacy," *Information Sciences*, vol. 560, no. 347–369, 2021.
- [15] S. Chen, A. Fu, S. Yu, H. Ke, and M. S. Dp-qic, "A differential privacy scheme based on quasi-identifier classification for big data publication," *Soft Computing*, vol. 25, no. 3, 2021.
- [16] Z. Gu, G. Zhang, and C. Yang, "Multi-party high-dimensional related data publishing via probabilistic principal component analysis and differential privacy," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, W. Shi, X. Chen, and KK. R. Choo, Eds., Springer International Publishing, Cham, Switzerland, pp. 117–131, 2022.
- [17] K. Nissim, F. D. Mcsherry, C. Dwork, and A. L. Blum, "Practical privacy: the sulq framework," in *Proceedings of the Twenty-Fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Baltimore, Maryland, USA, June, 2005.
- [18] K. Chaudhuri, A. D. Sarwate, and K. Sinha, "A near-optimal algorithm for differentially-private principal components," *Journal of Machine Learning Research*, vol. 14, pp. 2905–2943, 2013.
- [19] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Privbayes: private data release via bayesian networks," *ACM Transactions on Database Systems*, vol. 42, no. 4, pp. 1–41, 2014.
- [20] C. Rui, X. Qian, Z. Yu, and J. Xu, "Differentially private high-dimensional data publication via sampling-based inference," in *Proceedings of the 21th ACM SIGKDD International Conference*, Sydney, Australia, August, 2015.
- [21] X. Zhang, L. Chen, K. Jin, and X. Meng, "Private High-Dimensional Data Publication with junction Tree," *Journal of Computer Research and Development*, vol. 55, no. 12, 2018.
- [22] C. Xu, J. Ren, Y. Zhang, Z. Qin, and K. Ren, "Dppro: Differentially Private High-Dimensional Data Release via Random Projection," *IEEE Transactions on Information Forensics and Security*, vol. 1299 pages, 2017.
- [23] W. Zhang, J. Zhao, F. Wei, and Y. Chen, "Differentially private high-dimensional data publication via Markov network," *ICST Transactions on Security and Safety*, vol. 6, no. 19, Article ID 159626, 2019.
- [24] J. Ge, Z. Wang, M. Wang, and L. Han, *Minimax-optimal Privacy-Preserving Sparse Pca in Distributed Systems*, in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, Lanzarote, Canary Islands, April, 2018.
- [25] S. Wang and J. M. Chang, "Differentially private principal component analysis over horizontally partitioned data," in *In Proceedings of the 2018 IEEE Conference on Dependable and Secure Computing (DSC)*, Kaohsiung, Taiwan, December, 2018.
- [26] H. Imtiaz and A. D. Sarwate, "Differentially private distributed principal component analysis," in *Proceedings of the ICASSP 2018-2018 IEEE International Conference on*

- Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, April, 2018.
- [27] D. Alhadidi, N. Mohammed, B. Fung, and M. Debbabi, *Secure Distributed Framework for Achieving ϵ -differential Privacy*. Springer, Berlin, Heidelberg, 2012.
 - [28] X. Cheng, P. Tang, S. Su, R. Chen, Z. Wu, and B. Zhu, "Multi-party high-dimensional data publishing under differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 1–1, 2019.
 - [29] R. Wang, B. Fung, Y. Zhu, and Q. Peng, "Differentially private data publishing for arbitrarily partitioned data," *Information Sciences*, vol. 553, no. 10, 2020.
 - [30] C. Dwork, K. Kenthapadi, M. Frank, I. Mironov, and M. Naor, *Our Data, Ourselves: Privacy via Distributed Noise Generation*, DBLP, Trier, Germany, 2006.
 - [31] Cynthia, A. Dwork, and Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, 2013.
 - [32] H. Imtiaz and A. D. Sarwate, "Distributed differentially-private algorithms for matrix and tensor factorization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, pp. 1449–1464, 2018.

Research Article

An Edge Cloud Data Integrity Protection Scheme Based on Blockchain

Weihua Duan ¹, Yu Jiang ¹, Xiaolong Xu ^{1,2}, Ziming Zhang ¹ and Guanpei Liu ¹

¹Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

²School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

Correspondence should be addressed to Xiaolong Xu; xuxl@njupt.edu.cn

Received 1 January 2022; Revised 20 March 2022; Accepted 1 April 2022; Published 22 April 2022

Academic Editor: Xin-Yi Huang

Copyright © 2022 Weihua Duan 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 publicly accessible feature of edge servers leads to the threat of malicious access to the data stored on the server and a series of security problems such as the leakage of user data privacy and the destruction of integrity. Data custody causes the separation of user ownership and management rights and brings potential security risks of data theft and destruction. Among them, for the integrity of the data uploaded by the terminal, the current protection mechanism mostly verifies the identity of the visitor or encrypts the data, but the role of verification is mostly assumed by the server, and it is impossible to avoid the collusion of edge servers with malicious intruders. In this paper, a distributed virtual machine agent (VMA) is designed and implemented, an edge cloud data integrity monitoring framework is built, and the verification protocol based on blockchain is proposed, which achieves trusted verification without relying on a trusted third party. Also, a prototype system of edge cloud data integrity protection based on blockchain is constructed to prevent data corruption. The results of security proof and experimental verification show that the mechanism based on blockchain technology can defend against three attacks of cloud service providers, has superior computation, and reduces the storage costs to protect the integrity of user data.

1. Introduction

Cloud computing [1–3] is a computing model that uses the Internet anytime, anywhere, and quickly access shared resource pools (such as computing facilities, storage devices, and applications) in the form of on-demand services to provide users. However, with the increasing number of devices in the network and the exponential growth of generated data, cloud computing is difficult to handle massive amounts of business, resulting in a large amount of time delay, providing users with unsatisfied service experience. Therefore, edge computing [3, 4] is proposed to provide services that are closed to IoT devices with a short delay. An edge cloud [2, 5, 6] consists of edge servers located close so that it can use server collaboration to complete tasks from the IoT devices more efficiently and realize the real-time need.

Edge devices depending on distributed edge servers that belong to different enterprises and suppliers are scattered

and their processing capabilities are limited. Due to the heterogeneity of devices, most authentication and communication encryption technologies are not suitable for networking, causing data to suffer a huge threat, and data integrity cannot be guaranteed. Data custody causes the separation of user ownership and management rights and brings potential security risks of data theft and destruction.

On the one hand, the Edge Cloud Service Provider (ECSP) may either privately delete user data or deliberately conceal accidental data destruction for maintaining its own reputation. On the other hand, edge servers may be maliciously attacked, resulting in data destruction and loss of sensitive data. Edge cloud data integrity protection mechanism can ensure that data are stored in the edge cloud unmistakably and can immediately warn and reduce losses when data are illegally tampered.

The traditional methods of data integrity verification mainly focus on the integrity of local disk data and database data and adopt the scheme of integrity verification such as

digital signature, message verification code, and digital watermarking [7], but the current data integrity research is mainly for data in cloud computing. Yan et al. [8] adopted a combination monitor of inside and outside in a virtual machine, which proposed the security proposal for a virtual machine computing environment. The method provides the monitoring architecture of the virtual machine to ensure trusted computing but increases the hardware cost of the cloud platform and extra computing costs, especially since the architecture is not scalable. Erway et al. [9] improved a PDP scheme based on a Rank-based Authenticated Skip List (RASL). In 2015, Tian et al. [10] provided a dynamic data integrity mechanism based on the distributed hash table (DHT), which supports public authentication. However, the verification is conducted by the third-party auditors in these schemes. Users may be deceived by fake verification and collusion with CSP. Xu et al. [11] proposed a data validation algorithm to defend against spoofing attacks from untrusted validation results, which improved the reliability of validation results in 2017. However, dual validation evidence was introduced to cross-validate the validation results, which added computation and storage costs.

In edge computing, each server can process tasks for users independently and save the uploaded data. When the tasks submitted are difficult to handle, they will be submitted to the cloud center. This kind of scene realizes the decentralized scene at the edge and refuses the centralized manager.

The research on data integrity protection of edge cloud has made some progress in recent years. Wang et al. [12] achieved a balance through a balanced truth discovery method and the proposed data privacy enhancement technology and used these technologies to interact with IoT devices and edge servers. Chadwick et al. [13] proposed a framework that allows the secret sharing of cyber threat information (CTI) among partners for analysis. Li et al. [14] proposed a privacy protection data aggregation scheme for mobile edge computing-assisted IoT applications. The data privacy of the terminal device is guaranteed, and source authentication and integrity are also provided. Wang et al. [15] proposed an edge-based data collection model, in which the raw data from the wireless sensor network (WSN) is differentially processed by an algorithm on the edge server for privacy calculations.

Recently, Blockchain technology [16–18] has become popular worldwide. The blockchain guarantees the consistency of the data between nodes by a consensus algorithm and ensures data security by the encryption algorithm. In addition, formed by the timestamp and hash algorithm, the chained structure produces a series of technical features, such as openness, transparency, authentication, and tamper resistance [19]. The theory of smart contracts [20], firstly proposed by Nick Szabo, refers to a computer program which conducts terms of contract automatically. Blockchain technology, with a characteristic of multistorage, multiparty calculation, transparent rules, and tamper-resistant features, provides a reliable record carrier and execution environment for the smart contract.

This paper proposes a distributed virtual machine proxy architecture and a multitenant jointly safeguards the private chain based on the blockchain in the edge cloud and designs an edge cloud data integrity protection mechanism. The mechanism is oriented to the incredible edge cloud system and reaches a consensus agreement to complete credible integrity verification through the exchange of information. The main contributions of this paper are as follows:

- 1 Mobile Agent is used to deploy the distributed model of the virtual machine agent in the edge cloud. Virtual machine agents of multitenants cooperate to ensure data credible verification. The virtual machine agent mechanism completes not only reliable storage, monitoring, and verification of cloud data tasks but also is necessary to build a data integrity verification mechanism based on blockchain.
- 2 A blockchain integrity monitoring framework is built through the model of a virtual machine agent. This paper uses the Merkle Hash Tree to generate the unique value corresponding to data and monitor data changes with a smart contract in the blockchain for sending timely warnings of data destruction to the owner. In addition, the “challenge-response” model is used to construct the scheme of edge cloud data integrity verification.
- 3 This paper constructs and implements a prototype system of edge cloud data integrity protection based on blockchain and applies the integrity monitoring scheme based on virtual machine agents and the integrity verification scheme based on blockchain. After security certification analysis, the mechanism can defend against three kinds of attacks by edge cloud service providers and has a better performance compared with existing solutions.

The rest of the paper is organized as follows. Section 2 introduces the related work about the integrity verification mechanism based on the third party and blockchain technology. Section 3 puts forward blockchain architecture for cloud data integrity based on distributed virtual machine agents. Section 4 presents safety certification according to the scheme. Section 5 perfects experimental verification and performance analysis. Section 6 realizes the prototype system. Finally, section 7 summarizes and evaluates all of the work and points out the direction of further study.

2. Related Work

Data integrity verification in the edge computing environment has attracted more and more scholars' attention. Wang et al. [12] proposed a scheme that maintains a balance in three aspects, including user privacy, data integrity in edge-assisted IoT devices, and computing cost. Through the identity verification algorithm based on biometric ECC, the privacy participation of IoT users is authenticated during the truth discovery process, not only reducing the overall computing cost of the IoT equipment but also limiting the communication between the user equipment and the edge

server. Chadwick et al. [13] proposed a five-level trust model based on cloud edge data sharing infrastructure. Data owners can choose the appropriate level of trust and CTI data cleaning methods, from plain text to anonymization/pseudonymization to homomorphic encryption, so that CTI data can be manipulated before sharing it for analysis. Li et al. [14] proposed a privacy protection data aggregation scheme for mobile edge computing-assisted IoT applications. In the proposed model, there are three participants, namely terminal devices, edge servers, and public cloud centers. The data generated by the terminal device is encrypted and transmitted to the edge server. The edge server aggregates the data of the terminal device and submits the aggregated data to the public cloud center. Finally, the aggregated plaintext data can be recovered by the private key of the public cloud center.

Blockchain technology has been widely used in cryptocurrency since the emergence of Bitcoin [16]. IBM Blockchain [21] offers developers opportunities to develop their own applications based on the Hyperledger Fabric, which has been widely used in the financial industry, insurance industry, food safety, and so on. For instance, IBM and Wal-Mart cooperate to guarantee food safety by food traceability. Azure Blockchain [22] allows customers to quickly configure and deploy consortium chain networks, which supports lightweight development and testing workloads and even large-scale production blockchain deployment. The blockchain can shorten development time and costs through the cloud services required for application development. Amazon Managed Blockchain [23], which helps users use Ethereum and Hyperledger Fabric to create and manage a scalable blockchain network, eliminating the need to create a network, and continuously monitoring the blockchain network to quickly adapt to changes for application requirements has been used in many fields, such as financial and trade alliances. All parties in the blockchain can trade electronically and process trade-related paperwork without central trust.

Wang et al. [15] proposed an edge-based data collection model, in which the raw data from the wireless sensor network (WSN) is differentially processed by an algorithm on the edge server for privacy calculations. A small amount of core data are stored on the edge and local servers, while the rest is transmitted to the cloud for storage. Tian et al. [24] proposed an effective privacy protection authentication framework. By using a lightweight online/offline signature design, authentication efficiency is guaranteed when deployed on small drones with limited resources. Considering the high mobility of drones, a predictive authentication method is studied using mobile edge computing (MEC) in the framework to further reduce the cost of identity verification for potential identity verification activities. In addition, Wang et al. [25] designed a service selection method which selects corresponding credible and reliable service providers based on trust evaluation and recording standards, which has obvious advantages in terms of concise trust management, convenient service search, and accurate service matching. Establishing and maintaining a unified and trusted environment based on edge computing can detect

malicious service providers and service consumers in a timely manner, filter out false information, and recommend trusted service providers.

Yue et al. [26] proposed a blockchain-based framework without third-party auditors for data integrity verification in distributed edge cloud storage (ECS) scenarios. In the framework, a Merkle tree with random challenge numbers is used for data integrity verification, and different Merkle tree structures are analyzed to optimize system performance. In view of the problems of limited resources and high real-time requirements, sampling verification is further proposed, and reasonable sampling strategies are formulated to make sampling verification more effective.

Bonnah et al. [27] proposed a completely decentralized method to solve the untrustworthy problem of trusted parties by eliminating the public trusted entity in the network framework. Within the proposed framework, authenticated users do not have to log in to each service provider to be authenticated to access services or resources.

Ma et al. [28] proposed a blockchain-based edge computing trusted data management scheme for dishonest data. They proposed a flexible and configurable blockchain architecture, including mutual authentication protocols, flexible consensus and smart contracts, block and transaction data management, blockchain node management, and deployment. Before data storage in the blockchain system, a user-defined encryption method for sensitive data is designed, and conditional access and decryption queries for protected blockchain data and transactions from the blockchain system are designed.

Kang et al. [29] used blockchain and smart contract technology to realize secure data storage and sharing in the vehicle edge network. These technologies effectively prevent unauthorized data sharing. It also proposed a reputation-based data sharing program to ensure high-quality data sharing between vehicles. A three-weight subjective logic model is used to accurately manage the reputation of the vehicle.

Gai et al. [30] proposed a new method that combines the IoT with edge computing and blockchain. The proposed model is designed for a scalable and controllable IoT system, making full use of the advantages of edge computing and blockchain to establish a privacy protection mechanism while taking into account other constraints, such as energy costs.

In order to efficiently audit the integrity of application vendors' cached data, Li et al. [31] analyzed the threat model and audit objectives and proposed a lightweight sampling-based probabilistic method, including a variable Merkle hash tree. A new data structure of variable Merkle hash trees is designed to implement integrity proofs for generating copies of these data during audits.

Tong et al. [32] proposed two integrity checking protocols for mobile edge computing, checking the data integrity at the edge based on the concept of provable data ownership and proprietary information retrieval techniques. Liu et al. [33] modeled data failures by classifying them into format failures, time series failures, and value failures and proposed several heuristic rules for the detection and

isolation of data failures. Aujla et al. [34] designed a blockchain-based secure data processing framework for the Internet of Vehicles in the edge environment, including a container-based optimal data processing solution and a blockchain-based data integrity management solution, which can minimize link interruptions.

3. Study on Edge Cloud Data Integrity Protection Mechanism

Aiming at the problem of the untrustworthiness of data integrity verification in edge cloud, this paper designs a distributed virtual machine agent model in edge cloud combined with characteristics of blockchain technology and achieves consensus through multinode collaboration to complete the credible verification of edge cloud data. The design focuses on solving three problems. First, the integrity verification by a virtual machine agents prevents data leakage to third-party auditors; Second, credible proof on the blockchain ensures the credible validation results. Third, blockchain monitors the entire lifecycle of user data to ensure that data is not illegally tampered with.

3.1. Distributed Virtual Machine Agent Model. Virtual machine node is divided into two categories in function, including virtual machine agent (VMA) Node and storage node. When the user submits a storage task, the data are preprocessed by the VMA node, which is responsible for selecting the appropriate storage node, and after all the storage is done, the VMA node returns the result to the user. Different from cloud computing, edge computing is to sink resources near the data source and process the user's tasks close to the device. The data does not have to be uploaded to the data center, thus reducing the pressure on network bandwidth. The edge cloud can form a server cluster of edge servers with similar geographical locations and use server cooperation to complete tasks at the edge, reducing the delay of data transmission. Therefore, compared to cloud computing, edge computing is a highly decentralized distributed computing architecture.

For the application of complex services in the edge cloud distributed environment, and to enhance the portability of the model, the paper refers to the cloud environment and uses the Mobile Agent [35, 36] (MA) technology. MA is an agent in the network which performs specific processing in distributed problems. In the standard of FIPA [37] (Foundation of Intelligent Physical Agents), Agency is a container for carrying MA, and it may carry a plurality of MA and provide an operating environment for performing any MA. An agency can carry a number of MA, and MA can be run in the agency. Therefore, the running agency in the node can complete the model deployment of distributed virtual machines agent. Figure 1 is a node structure of the user in the edge cloud.

Definition 1. VMA node, proxy node in edge cloud, logically unique, is responsible for acting on behalf of the user to perform various tasks with high computing power.

Definition 2. Storage node, storage for edge data, not unique. All storage nodes consist of Interplanetary File System [38] (Interplanetary File System, IPFS) cluster which is responsible for storing massive data with lower computing power.

After the deployment of the virtual machine agent model, the paper uses blockchain technology to union nodes, which aims at achieving a consensus agreement through the exchange of information to ensure chain data is open, transparent, tamper-resistant, and traceable. Blockchain is divided into a public chain, private chain, and consortium chain in accordance with the authority of the consensus process. This paper adopts a private chain, giving cloud tenants the privilege to read and write, preventing outside interference in the consensus process. In addition, in order to prevent malicious attacks, we take tokens way to produce a transaction. Each node has a certain initial token, and every deal needs to consume tokens. Once successfully obtained the right to package block, edge nodes will receive some token reward so as to encourage tenants open owner VMA to participate consensus process.

As shown in Figure 2, this paper introduces a distributed virtual machine agent model to build a basic protection framework for edge cloud data integrity. When the user submits the storage task, the data is first uploaded to the VMA node, and after the preprocessing, a transaction is generated into the buffer pool. The transaction stores the evidence of data integrity verification. VMA nodes perform polling, querying the transaction that has not been confirmed in the buffer pool and once found, the VMA is trying to verify the legitimacy of the transaction and packages to form a group of the block legitimate transactions.

3.2. Workflow. The aim of the section is to build a blockchain network through interaction with the VMA for the preparation of integrity protection.

3.2.1. Connection and Synchronization. Blockchain network is based on P2P protocol and there are no central authority nodes. Each node can broadcast routing, discover new nodes, and allow dynamic legitimate nodes to join or quit. The underlying blockchain platform is not limited to Ethernet Square, Ethermint, Fabric, and so on, as long as there are many functions such as account inquiries, transactions, contracts, and other operational intelligence functions.

Step 1. First, the ECSP deploys a blockchain network in the edge cloud and runs the initial file to generate a first block (block Genesis), waiting for the VMA of tenants.

Step 2. Once joining the blockchain network, user's VMA verifies itself whether the data block is the latest in the blockchain network or not. If yes, VMA monitors data broadcasting in the network. Otherwise, block data synchronization neighbor nodes. Then use the public key to verify the legitimacy of transactions.

Step 3. When listening to new transactions and blocks, VMA verifies the signatures of those transactions and

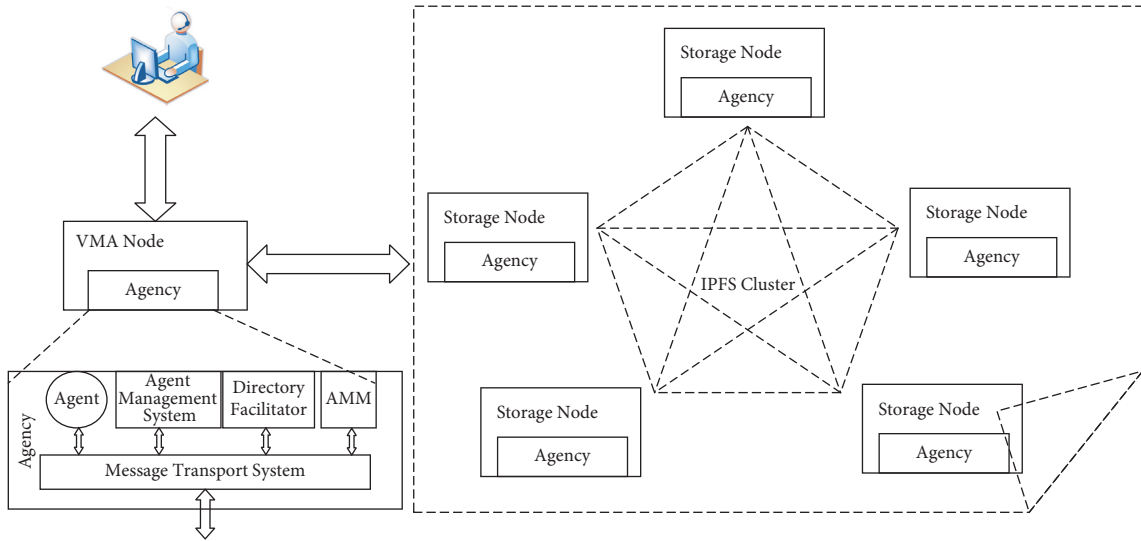


FIGURE 1: Storage node in edge cloud architecture.

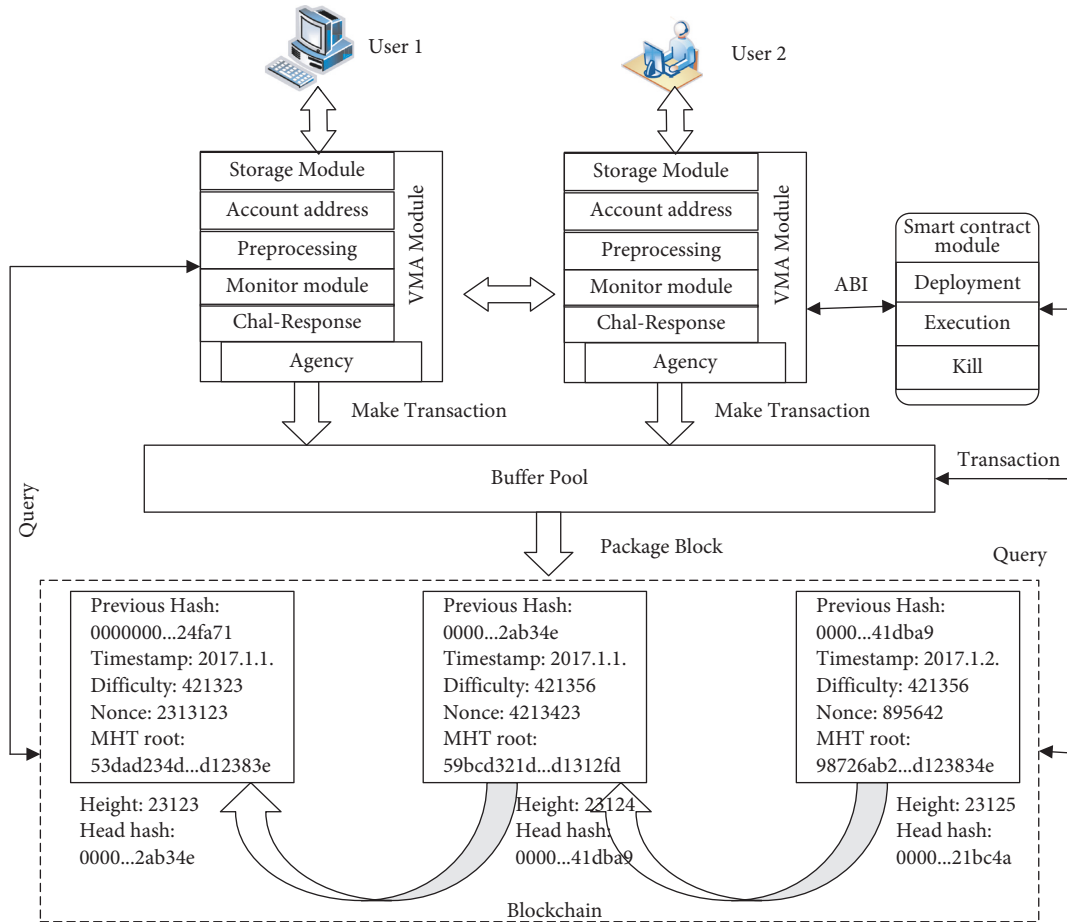


FIGURE 2: Blockchain architecture for edge cloud data integrity.

blocks. If signatures are valid, VMA processes and forwards them by the consensus module to prevent invalid data from propagating.

3.2.2. Storage. Once users upload files, VMA preprocesses the file and uses IPFS cluster storage. Based on the contents alternate instead of domain, IPFS uses http browser to search files, firstly locates the server, and then uses the pathname to find files on the server. Specific steps are as follows:

Step 1. When file is added to an IPFS node, a unique encrypted hash fingerprint is calculated from the file contents, ensuring that the value always only indicates the contents of the file. Even if you modify a bit of data in a file, the hash fingerprint will be completely different.

Step 2. The next step, users query hashing by the distributed hash table in the IPFS distributed network, which uses a consistent hash function to unify the machine's IP address and data, and quickly (The network only needs 20 hops in a system with 10,000,000 nodes) find the node that owns the data, retrieve the data, and use hashes to verify if this is the correct data.

3.2.3. Deployment of Smart Contract. Smart contracts have decentralized computations and storage based on blockchain. After the blockchain network is established, smart contracts will be deployed.

Step 1. After writing a smart contract, users use the browser compiler Remix to compile code into binary code.

Step 2. Users consume some tokens to deploy the compiled contract to the network, access to contract address of blockchain, and Application Binary Interface (ABI). ABI is a binary representation of the interface contract.

Step 3. When the user uploads the file, the IPFS address and Merkel Hash Tree [39] (MHT) root hash value will be obtained by preprocessing the file, and they are stored as a key-value pair in the data structure of the map by invoking smart contract by contract address and the ABI.

Step 4. When the user checks the file, users use the IPFS address of the file as the key to obtaining the MHT root hash value in the smart contract for comparison.

3.2.4. Destroy. Once deciding to delete VMA, tenants first call the kill function in smart contract for deleting the contract data and recovering the remaining tokens. And then, VMA starts the self-destruction of the module and the data will be rewritten overlay.

3.3. Blockchain Based Integrity Protection Mechanism. Based on the VMA architecture, the private chain is created and jointly safeguarded by the tenants in the edge cloud. The information can be traced back, tamper-resistant, and the

trusted execution in the blockchain and smart contract. Therefore, this paper designs a data integrity monitoring program and blockchain integrity verification protocol based on the "challenge-response" model. The protocol is also based on the bilinear mapping of BLS [37] (Boneh-Lynn-Shacham) short signature verification [39, 40] and the mechanism is divided into three parts.

3.3.1. Pretreatment Stage. Get big primes p , $p \in Z_p$, set G_1, G_2 is Multiplication cycle group of prime number p , g_1 is the generator of G_1 , g_2 is the generator of G_2 . There is a bilinear map, $\ell: G_1 \times G_1 \rightarrow G_2$. Randomly select $a, x \in Z_p$, $u = g_1^a$. The user generates a key pair $\{SK = \{a, sk\}, PK = \{g_1, u, pk\}\}$ locally, where the private key $sk = x$, public key $pk: v = g_2^x$.

Step 1. The user sends a request to connect the corresponding virtual machine agent. The VMA receives the user's request and then verifies whether it is valid or not. If the request is valid, the VMA will agree to connect. If not, a connection refusal response will be returned.

Step 2. Users upload files to VMA and VMA initializes data files. Firstly, the data information F is Partitioned into block $F = \{m_1, \dots, m_i, \dots, m_n\}$. Secondly, each block is divided into a segment, that is $m_i = \{m_{i,1}, \dots, m_{i,j}, \dots, m_{i,k}\}$, $1 \leq j \leq k$. Finally, call the tag generation algorithm for each data block. Generate a digital signature as follows:

$$\sigma_i = \left(H(b_i \| t_i) \cdot \prod_{j=1}^k g_1^{(a_j) \cdot h(m_{i,j})} \right)^x = \left(H(b_i \| t_i) \cdot \prod_{j=1}^k u_j^{h(m_{i,j})} \right)^x \quad (1)$$

where H and h are hash function. $H: \{0, 1\}^* \rightarrow G_1$, $h: \{0, 1\}^* \rightarrow Z_p$. $a_j \in Z_p, x \in Z_p$. The data segment number b_i , timestamp t_i , $1 \leq i \leq n$. $\Phi = \{(\sigma_i) | 1 \leq i \leq n\}$ is a data information file F tag set of data blocks, the tag is stored in the database of the virtual machine agent.

Step 3. VMA uploads data F to store in the IPFS cluster and returns the IPFS address $F_I d$, $F_I d$ is unique identifier of data.

3.3.2. Data Integrity Monitoring Stage. After preprocessing of file integrity verification, VMA stores the digital signature of the data block in the database and computes the MHT root hash value according to the digital signature. The root value is deployed to the blockchain by invoking smart contracts. MHT is a kind of binary tree, as shown in Figure 3 [39]. The data tag value is stored only at the leaf nodes. The nonleaf nodes are obtained by the hash operation after linking the values of the left and right subnodes. Finally, the root hash value represents the integrity of the whole file.

Through the MHT root node, the tampering of any data block is detected to ensure the integrity of the file without the participation of other nodes of MHT. Meanwhile, MHT has only been a directed branch from the measured node to the MHT root node path, which can confirm whether the node exists in the data block or not, for example, verifying whether

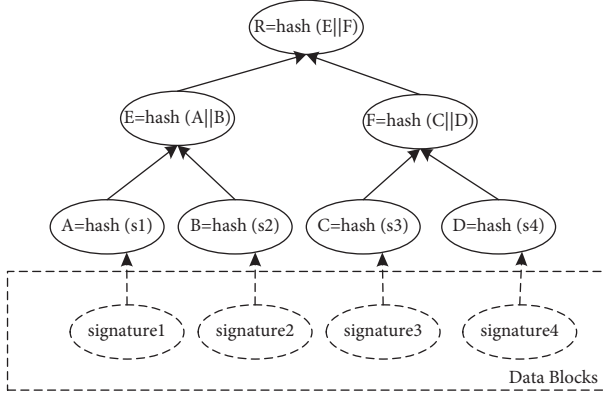


FIGURE 3: MHT structure diagram.

D is in the block according to the nodes C , E , and R . There are N data blocks in the MHT; hash computing is $2 \log_2 N$; it can verify whether the data block has been tampered with or not.

The process of implementing the integrity monitoring mechanism is as shown in Figure 4. The process is as follows: The user uploads the file to the VMA for preprocessing. On the one hand, the file is divided into blocks, the tag is stored in the database, and the data tag of the file block generates the MHT; On the other hand, storing the file in the IPFS cluster gets the address based on context. By invoking the smart contract to save key-value pairs, the blockchain will monitor the value whether the file is modified.

3.3.3. Edge Cloud Data Integrity Verification Stage. When users are concerned that the data has been tampered with, users only need to challenge the ECSP. According to the ECSP's response, users can know whether the data is complete.

Step 1. The user sends a request of data integrity verification for the file to be detected. The request includes the data block set $INDEX = \{i | dx_i | 1 \leq i \leq c, c \leq n\}$ and the corresponding random number set $R = \{r_i | i \in INDEX, r \in Z_p\}$.

Step 2. Firstly, according to the challenge request, the VMA node queries the IPFS cluster for the IPFS unique flag $F.I.d$. Secondly, the VMA node creates a MA to migrate to the storage node to obtain the corresponding evidence of the data block. Variable c represents the total challenge number of data blocks to be detected, n is the total number of data blocks in the data block set.

Step 3. Storage node obtains the corresponding data block $\sum_{i \in INDEX} h(m_{ij})$ by executing the MA task, returns the value to the VMA node, and calculates the total data block:

$$M = \sum_{j=0}^k \sum_{i \in INDEX} h(m_{ij}). \quad (2)$$

According to the VMA node stored u , VMA calculates the total digital signature of the data block:

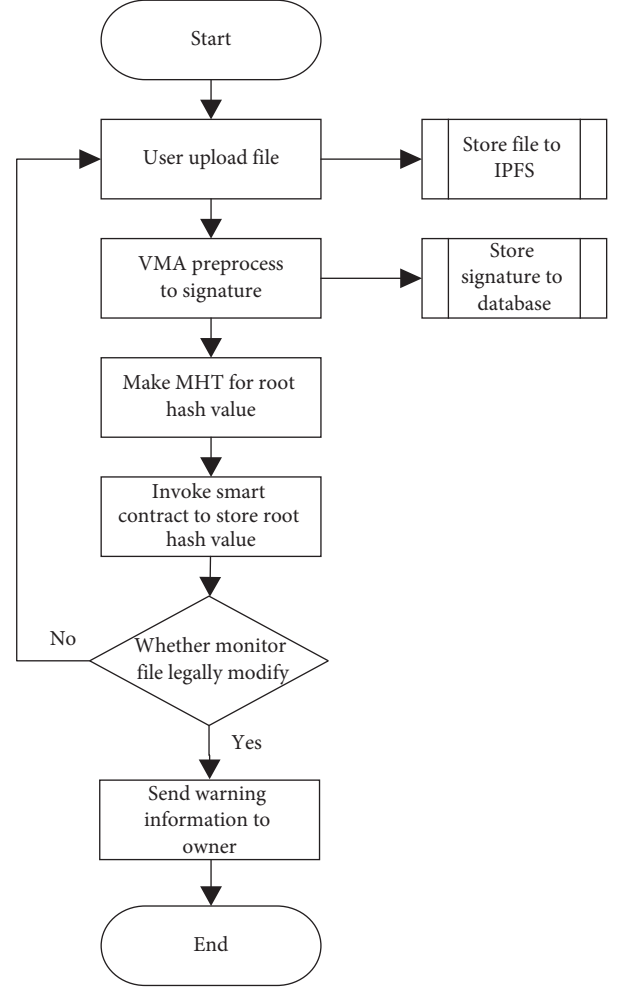


FIGURE 4: Integrity monitoring flow chart.

$$D = \prod_{j=0}^k u_j^{h(m)} = \prod_{j=0}^k u_j^{\sum_{i \in INDEX} \sum_{j=0}^k h(m_{ij})}. \quad (3)$$

Step 4 VMA reads the challenge data block tag value from its own database. And it then calculates the hash value of the corresponding challenge block number:

$$\begin{aligned} T &= \prod_{i \in INDEX} \sigma_i^{r_i}, \\ B &= \prod_{i \in INDEX} H(b_i \| t_i)^{r_i}. \end{aligned} \quad (4)$$

It generates evidence proof $= \{D, B, T\}$ and calculates:

$$l(B, v) \cdot l(D, v) \triangleq l(T, g_2). \quad (5)$$

If (5) holds, the supporting documents are complete.

Step 5. The user will receive the verification result of VMA and get the file MHT root hash value. If both values are equal, the verification result is credible.

The integrity verification stage is shown in Figure 5: some data blocks are randomly extracted by users. Users send a challenge to ECSP by VMA node. Firstly, the IPFS

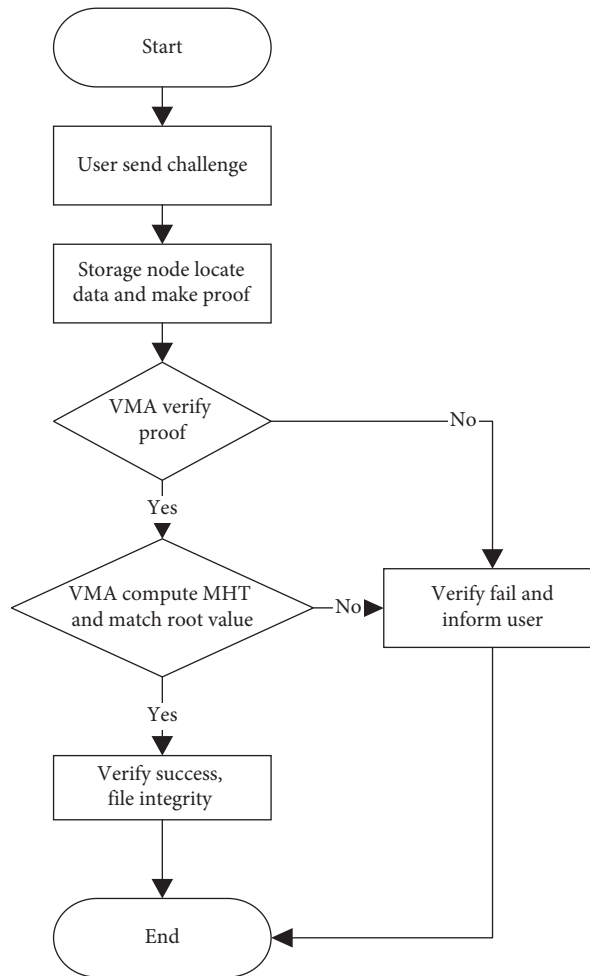


FIGURE 5: Integrity verification flow chart.

cluster fixes positions according to the challenge data block and generates the evidence back to the VMA. VMA verifies (5) and calculates the validity of the evidence. If valid, the second step validation will be performed to calculate whether the challenge block exists and whether the root hash value is consistent through the MHT. If consistent, the document is integrity, or the file is damaged.

4. Safety Certification

4.1. Analysis of Blockchain Integrity Monitoring Scheme. Based on the blockchain edge cloud data integrity mechanism, the following is considered for the normal modification and illegal tampering of validation results.

4.1.1. Attack Mode Analysis. ECSP attacks are divided into three levels.

First-level attacker: The attacker can break the security protection of the virtual machine agent and obtain the access control rights of the user, such as unauthorized users.

Second-level attacker: The attacker not only controls the virtual machine agent but also obtains the user's blockchain account address and key. The attacker invokes the intelligent

contract interface and modifies the MHT hash value saved in the blockchain.

Third-level attacker: The attack value is illegally invaded into the IPFS cluster of the distributed storage system to tamper with and destroy the data. For example, administrators have the highest authority on data management, and if they are curious about user's data, they have direct access to the user's data at the storage node.

4.1.2. For Integrity Monitoring Mechanism Analysis of Normal and Illegal Tampering Validation Results.

Normal modification: The user node sends an access request. Firstly, the VMA reads the data from the database and obtains the hash fingerprint of the corresponding data. Secondly, VMA obtains the data from the IPFS cluster and transmits the data to the user node. After the user modifies the data and invokes the smart contract, the VMA collects the affected data and generates a new digital label. Finally, VMA calls the smart contract interface to save the new MHT root hash to generate a new transaction. IPFS will update the database information.

Illegal tampering: As shown in Figure 6, if a third-level attacker attacks the storage data in the IPFS cluster, this

method can only select the data block number randomly from the sample integrity verification by the user. From the corresponding hash value found in the database, the file block is obtained by using IPFS, and the integrity verification operation is performed by the virtual machine agent.

If attacked by a first-level attacker, the data is illegally tampered with and the hash value and the digital signature corresponding to the data block in the database of the VMA change, so the root hash value generated by the MHT is also changed. Different from the value saved by the smart contract in the blockchain, the tampering failed.

If attacked by a second-level attacker, the attack value not only controls the virtual machine agent but also obtains the user's blockchain account and key and attempts to invoke the smart contract interface to modify the MHT root hash of the file. If successful, the transaction record will be left and saved by the other tenants; if it fails, the user will be warned through the smart contract.

4.2. Certificate of Integrity Agreement. This section will analyze the security of the scheme and propose that the system model may be attacked by three kinds of attacks [41] to solve the possible threats.

Theorem 1. *Proof of equality is whether it is established; if established, the document is complete; otherwise, the document has been tampered with.*

The proof is given as follows:

$$\begin{aligned}
& \ell(B, v) \cdot \ell(D, v) \\
&= \ell\left(\prod_{i \in \text{IDX}} H(b_i \| t_i)^{r_i}, v\right) \cdot \ell\left(\prod_{j=0}^k u \sum_{i=0}^k i = 0^k \sum_{j \in \text{IDX}} h(m_{ij}), v\right), \\
&= \ell\left(\prod_{i \in \text{IDX}} H(b_i \| t_i)^{r_i}, v\right) \cdot \ell\left(u^{\sum_{i \in \text{IDX}} h(m_i)}, v\right), \\
&= \ell\left(\prod_{i \in \text{IDX}} H(b_i \| t_i)^{r_i} \cdot u^{\sum_{i \in \text{IDX}} h(m_i)}, v\right), \\
&= \ell\left(\prod_{i \in \text{IDX}} \left(H(b_i \| t_i) \cdot u^{h(m_i)}\right)^{r_i}, g_2^x\right), \\
&= \ell\left(\prod_{i \in \text{IDX}} \left(H(b_i \| t_i) \cdot u^{h(m_i)}\right)^{x r_i}, g_2\right), \\
&= \ell\left(\prod_{i \in \text{IDX}} t_i^{r_i}, g_2\right), \\
&= \ell(T, g_2).
\end{aligned} \tag{6}$$

Theorem 2. *Forge attacks. If the data owner reuses a certain secret value for different versions of data when generating a signature, then in the storage node, the ECSP may forge the data signature of the data block to deceive the verifier.*

Proof. In the integrity verification mechanism, ECSP is not feasible to forge audit evidence in order to pass verification.

Game Definition: The user sends a challenge message to the storage node of the ECSP through the VMA:

$$\text{chal} = (\text{IDX} = \{i \mid dx_i, 1 \leq i \leq c, c \leq n\}, R = \{r_i \mid i \in \text{IDX}, r \in Z_p\}). \tag{7}$$

In order to verify (5), the ECSP should send based on the correct file, Audit evidence, but the ECSP constructed the evidence from the wrong data.

$$\begin{aligned}
& \text{proof} = \{D', B, T\}, \\
& D = \prod_{j=0}^k u_j^{h(M)} = \prod_{j=0}^k u_j^{\sum_{i \in \text{IDX}} h(m'_{ij})} = 0^k \sum_{i \in \text{IDX}} h(m'_{ij}), \\
& M' = \sum_{j=0}^k \sum_{i \in \text{IDX}} h(m'_{ij}).
\end{aligned} \tag{8}$$

□

Definition $h(\nabla m_i) = h(m'_i) - h(m_i)$, $i \in \text{IDX}$, at least one element is nonzero. If ECSP falsified evidence still passes VMA verification, ECSP wins the games, Otherwise, it fails.

Suppose ECSP won the game, according to the verification (5),

$$\ell(B, v) \cdot \ell(D', v) \triangleq \ell(T, g_2). \tag{9}$$

According to the dual mapping $u^{\sum_{i \in \text{IDX}} h(m_i)} = u^{\sum_{i \in \text{IDX}} h(m'_i)} \Rightarrow u^{\sum_{i \in \text{IDX}} h(\nabla m_i)} = 1$, G is a step for the multiplicative cyclic group of a prime number p . There are elements $c_1, c_2 \in G, \exists x \in \mathbb{Z}_p, c_2 = c_1^x$. Then $u = c_1^\alpha c_2^\beta \in G$, $u^{\sum_{i \in \text{IDX}} h(\Delta m_i)} = c_1^{\alpha \sum_{i \in \text{IDX}} h(\Delta m_i)} \cdot c_2^{\beta \sum_{i \in \text{IDX}} h(\Delta m_i)} = 1$, $\sum_{i \in \text{IDX}} h(\Delta m_i) \neq 0, \beta \neq 0, x = -\alpha/\beta$. The probability of $\beta = 0$ is $1/p$; then for the DL assumption $1 - 1/p$, the probability of solving contradicts the DL conjecture. Therefore, ECSP is proved as unforgeability.

Theorem 3. *Alternative attack, when the data block m_i or signature t_i is lost, the ECSP may replace the user's challenge with another valid data and data signature.*

Game Definition: The user sends a challenge message to the storage node of the ECSP through the VMA:

$$\begin{aligned}
& \text{chal} = (\text{IDX} = \{i \mid dx_i, 1 \leq i \leq c, c \leq n\}, \\
& R = \{r_i \mid i \in \text{IDX}, r \in Z_p\}).
\end{aligned} \tag{10}$$

In order to pass the verification equation above, the ECSP should send audit evidence $\text{proof} = \{D', B, T\}$ based on the correct document F . The ECSP constructs data block evidence

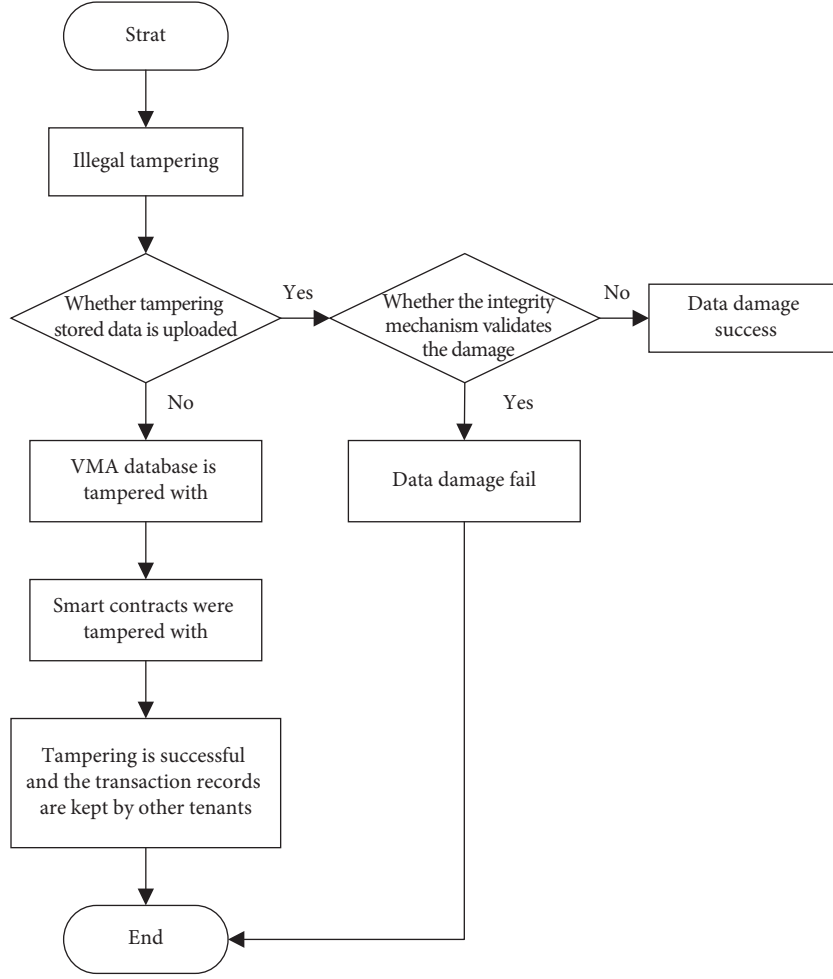


FIGURE 6: Illegal tampering with the flow chart.

$f - th(f \in \text{IDX})$ instead of $i - th(i \in \text{IDX})$. If it is proved by VMA and can still be verified, ECSP will win the game;

otherwise, it will fail. According to the properties of the bilinear map pair,

$$\begin{aligned}
 \ell(B, v) \cdot \ell(DI, v) &= \ell\left(\prod_{i \in \text{IDX}} H(b_i \| t_i)^{r_i}, v\right) \cdot \ell\left(\prod_{j=0}^k u_j^{\sum_{i \in \text{IDX} \& i \neq f} h(m_{ij}) + h(m_{fj})}, v\right), \\
 &= \ell\left(\prod_{i \in \text{IDX}} H(b_i \| t_i)^{r_i}, v\right) \cdot \ell\left(u^{\sum_{i \in \text{IDX} \& i \neq f} h(m_i)} \cdot u^{h(m_f)}, v\right), \\
 &= \ell\left(\prod_{i \in \text{IDX} \& i \neq f} \left(H(b_i \| t_i) \cdot u^{h(m_i)}\right)^{xr_i} \cdot \left(H(b_f \| t_f) \cdot u^{h(m_f)}\right)^{xr_f}, g_2\right).
 \end{aligned} \tag{11}$$

If the above formula is established, b_i is on behalf of the data section number, then $b_i = b_f$, $t_i = t_f$. Because the definition $f \neq i$, then $t_i \neq t_f$. Therefore $H(b_i \| t_i) \neq H(b_f \| t_f)$. ECSP replaces data signature failure.

Theorem 4. *Replay attacks; ECSP may not need to retrieve stored data; use the previous response to the evidence or other information to generate this evidence.*

Proof. The repeated attacks are defined as follows. The VMA sends a challenge request to the ECSP.

$$\text{chal} = (\text{IDX} = \{i \mid dx_i, 1 \leq i \leq c, c \leq n\}, R = \{r_i \mid i \in \text{IDX}, r \in \mathbb{Z}_p\}). \tag{12}$$

ECSP responds with an audit certificate proof = $\{D', B, T\}$. In the process of generating a proof, each data block $j - th(j \in \text{IDX})$ is replaced by the previous

information. This paper uses single quotes to separate the previous parameter from the correct parameter, for example: m'_j is the previous data block, m_j is the correct data block. If this proof is still validated by third-party audits, ECSP will resist replay attacks.

This proof is similar to Theorem 3, the same data block timestamps cannot be consistent. That is $H(b_j || t_j) \neq H(b_j || t'_j)$, ECSP will fail. \square

5. Prototype System

This paper uses advanced language (Solidity), which is designed to compile code that generates code that can run on the blockchain. The entire system is divided into three parts: Web client, VMA server, and blockchain API. As shown in Figure 7, a Web client mainly allows users to upload files, generate accounts address of blockchain and initiate challenges integrity verification operations. The VMA server can mainly preprocess files, respond to the challenge of integrity verification, establish MHT, and interact with the blockchain network by the blockchain API, such as account address generation, smart contract creation, and IPFS storage.

5.1. System Overall Process. Figure 8 shows three important functions of the prototype system: interacting with the account address generation, completing pretreatment, and verifying data integrity.

5.2. Function to Achieve

5.2.1. Web Client Implementation. Upload files: Upload files to the edge cloud VMA server, set the conditions of file division, and control the size of data blocks.

Download File: Save the file to the edge cloud IPFS cluster and obtain the source file based on the IPFS file address.

Initiate the challenge: The user selects the appropriate number sent to the VMA for integrity challenges according to the total number of files.

Register account address of blockchain: Provide user name and password to be completed by the VMA server registration.

5.2.2. VMA Server Implementation. Create a smart contract: Use the blockchain account address call ABI of the smart contract, and spend a certain token to generate a new contract address which is used to save the MHT root hash.

Register accounts address of blockchain: The account of each blockchain is composed of a pair of public and private keys, and the account address is 20-byte public key derived. The account uses public key encryption to sign the deal in order to send a secure authentication identity of the person in the blockchain network. The private key is encrypted with the password provided by the user. All blockchain operations are based on the address, and the same user can register multiple account addresses to prevent privacy disclosure.

Query Information on the blockchain network: According to the Transaction id or block number, the user

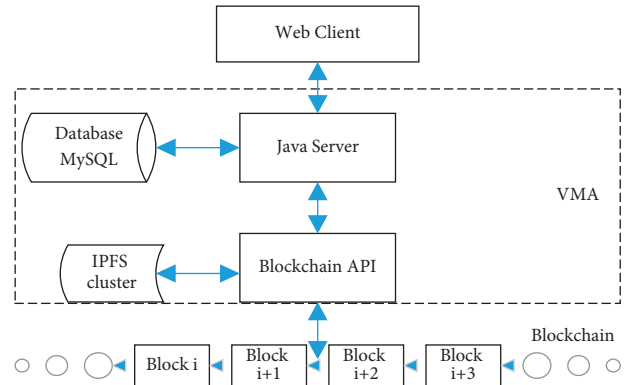


FIGURE 7: System frame diagram.

can query information of the blockchain to track changes to files and postaudit.

Preprocess file: The file is divided into data blocks according to user requirements and then generates a digital signature to calculate the MHT root hash value.

Verify challenge: According to the number of Web client challenges, VMA obtains the source data block from the ECSP, calculates the evidence, and verifies the data integrity.

5.2.3. Database Implementation. VMA database includes three tables, namely: (1) Fileinfo table, and the digital data signatures are being uploaded for integrity verification and MHT generation; (2) Public table, record public information and selected random number; (3) Users table, record the account address and transaction id and other related information.

6. Results and Discussion

6.1. Experimental Setup. The following contents will design experiments on this mechanism named Blockchain Proof of Data Possession (BPDP). Four virtual machines are used to simulate the VMA to form a blockchain network. Each virtual machine has the whole module for integrity verification. Users interact with the VMA through the web system. The integrity verification module uses JPBC (Java Pairing Based Cryptography) version 2.0.0. The elliptic curve uses MNT d159 curve. The basic domain size is 159, and the embedding degree is 6. The safety parameter selected experiment is 80 bit. The experiment in the system randomly generated a fixed size of the file F , and each experimental result takes the experimental average of 30 times.

6.2. Performance Analysis. At first, the integrity verification protocol is performed based on the accuracy of sample analysis. Assuming the total number of data blocks in the edge server is n if the number of error data blocks is e and corrupted data block ratio is $p_b = e/n$. Assuming that t is the ratio of the number of data blocks in each challenge to the total number n , then the probability of illegal tampering detected each time is as follows:

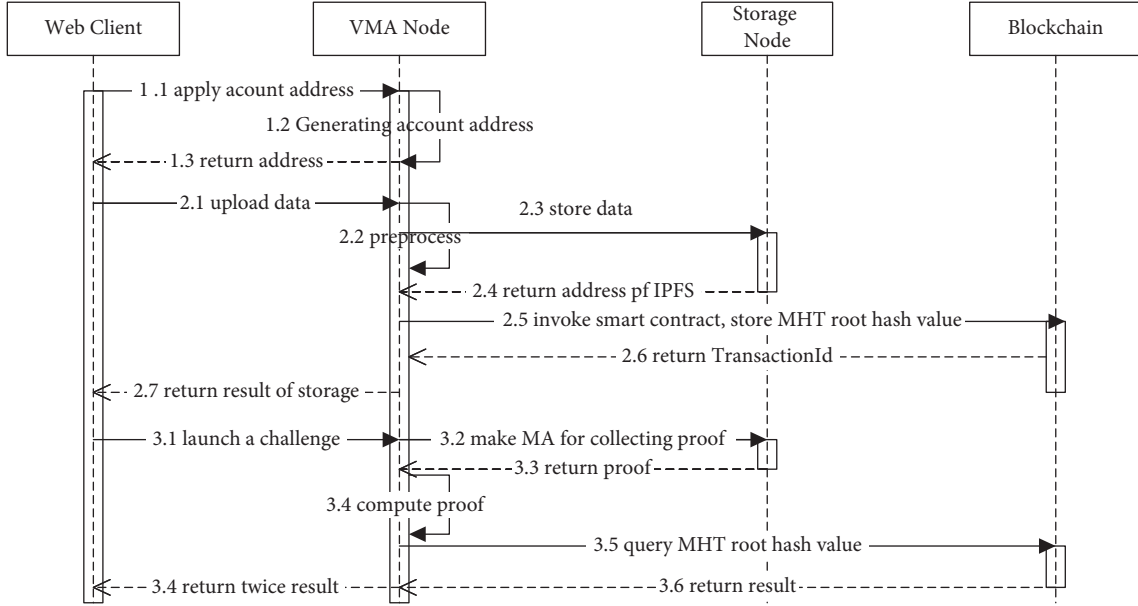


FIGURE 8: Prototype system time sequence diagram.

$$\begin{aligned}
 P &= P\{X \geq 1\} \\
 &= P\{X = 0\} \\
 &= 1 - \frac{n-e}{n} \cdot \frac{n-1-e}{n-1} \cdots \frac{n-t \cdot n-e}{n-t \cdot n} \geq 1 - \left(\frac{n-e}{n}\right)^{t \cdot n} \quad (13) \\
 &= 1 - (1 - p_b)^{t \cdot n}.
 \end{aligned}$$

As shown in Figure 9, if the number of error data blocks with the total number of data blocks ratio is 0.1%, the accuracy of 99%, and the total number of data blocks is 10,000, the number of challenge blocks is 4600. As shown in Figure 10, if the ratio of damage is 1%, then the number of challenge blocks is 460. Therefore all integrity protocols perform relatively poorly with less damage ratio. In the paper preprocessing, MHT is constructed to store the root hash of the file into the blockchain, which is twice used to ensure the file is not tampered with after sample integrity verification.

The security parameter selected in this paper is 80 bit, meaning $|p| = 160$. The storage cost of data signature is $n * p/8$, n is the number of data blocks. In order to achieve data dynamic operation, the establishment of an index hash table (IHTCost) spends storage cost is $n * (2 * p + 2 \log n)/8$. When the number of data segments is fixed, the larger the data segment and the smaller the number of formed data blocks will reduce the storage costs of the index hash table, as shown in Figure 11.

The most critical module of the prototype system is the edge cloud data integrity verification module. As shown in Figure 12, when the file block is too small, resulting in a dramatic increase in the number of file blocks and consumes longer pretreatment time. If the block is large, the number of

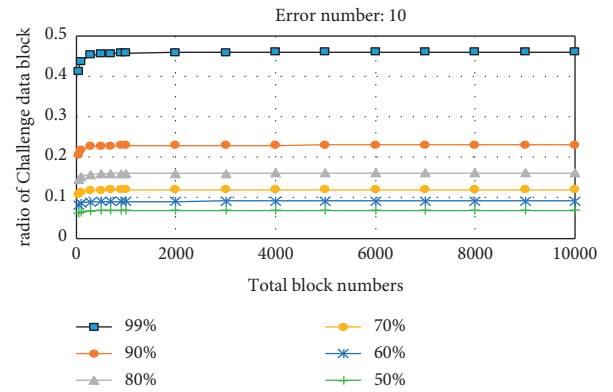


FIGURE 9: Challenge data block scale when the number of error block is 10.

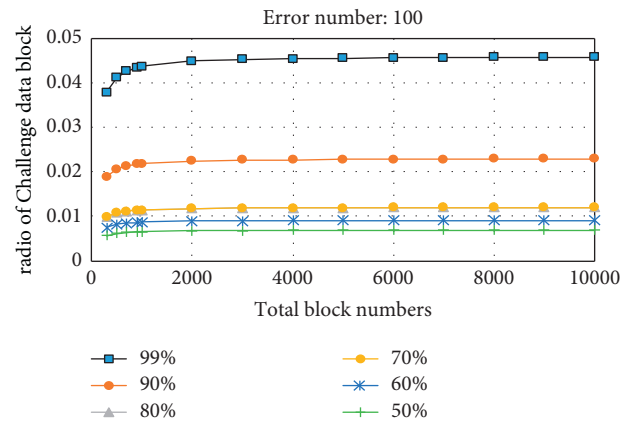


FIGURE 10: Challenge data block scale when the number of error block is 100.

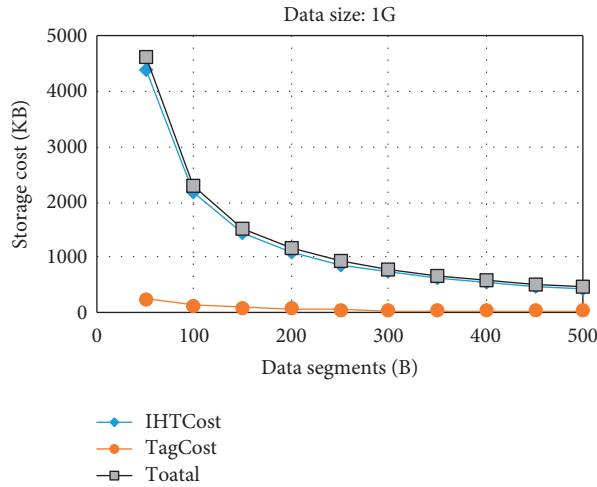


FIGURE 11: Challenge data block scale when the number of error blocks is 100.

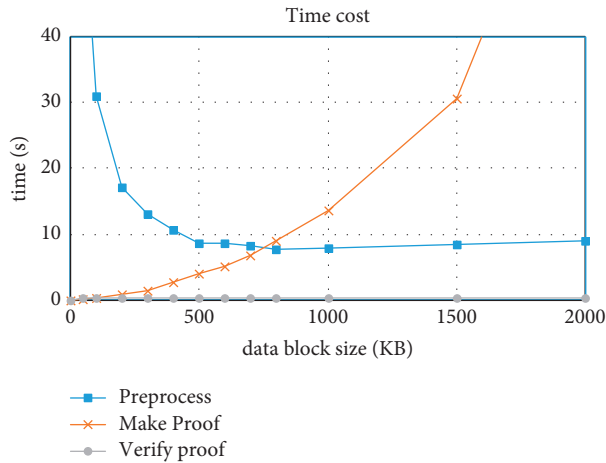


FIGURE 12: Prototype system time costs.

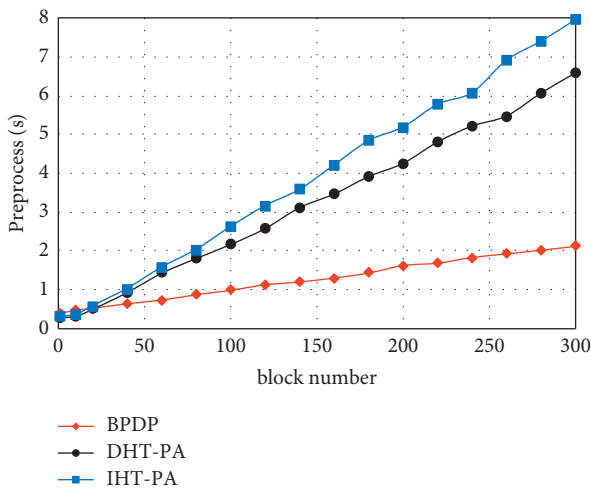


FIGURE 13: Verification time comparison.

data segments increases dramatically when the data block is divided into data segments resulting in an increase in the time for the generation of evidence. The file is set to 1G, and

12S completes a series of integrity validation when the number of data blocks is set reasonably.

Next, this paper analyzes the time costs of performing an integrity verification in DHT-PA [10], IHT-PA [42], and the BPDP. According to the accurate analysis of the verification, it is assumed the case when the error ratio is 1%. This paper selects the appropriate number of challenge blocks in order to achieve 99% accuracy. As shown in Figure 13, the experiment shows that the preprocessing time is proportional to the number of data blocks when processing the same size data block (50 KB). The result analyzes that the time costs of this paper are better than that of the same data block.

7. Conclusions

The above analysis shows that users store data on the edge cloud server and delegate the integrity verification of the remote data to the VMA so as to reduce the burden on users and eliminate the potential threats of third-party auditors. VMA itself is in the edge cloud and reaches a protocol consensus through information exchange in an unreliable, potentially threatening network, enabling trusted integrity

verification in an untrusted environment, protecting user data integrity, and preventing data from being illegitimate tampered with. In addition, the blockchain can save the interaction information of user and ECSP and record the nonrepudiation information which is manipulated by users' operations in the edge cloud environment so as to collect effective, reliable legal evidence to establish a perfect accountability mechanism. The next step will be to implement the access control of smart contracts according to the scheme, set access rights, and improve the control of user data, so as to better protect user data. It is hoped that the scheme can finally be put into production.

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 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 under Grant 62072255 and the Postgraduate Research and Practice Innovation Program of Jiangsu Province.

References

- [1] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [2] M. Du, Y. Wang, and K. Ye, "Algorithmics of cost-driven computation offloading in the edge-cloud environment," *IEEE Transactions on Computers*, vol. 69, no. 10, pp. 1519–1532, 2020.
- [3] K. Gai, J. Guo, and L. Zhu, "Blockchain meets cloud computing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2009–2030, 2020.
- [4] W. Shi, J. Cao, and Q. Zhang, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [5] E. El Haber, T. M. Nguyen, and C. Assi, "Joint optimization of computational cost and devices energy for task offloading in multi-tier edge-clouds," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3407–3421, 2019.
- [6] C. Li, J. Bai, and Y. Chen, "Resource and replica management strategy for optimizing financial cost and user experience in edge cloud computing system," *Information Sciences*, vol. 516, pp. 33–55, 2020.
- [7] Y. Fan, *Research on Cloud Data Integrity Verification and Data Recovery*, Chongqing University, Chongqing, China, 2016.
- [8] S. Yan, Y. Chen, and P. Liu, "Security protection mechanism of virtual machine computing environment under the cloud computing," *Journal on Communications*, vol. 36, no. 11, pp. 102–107, 2015.
- [9] C. C. Erway, A. K p c , C. Papamanthou, and T. Roberto, "Dynamic provable data possession," *ACM Transactions on Information and System Security*, vol. 17, no. 4, p. 15, 2015.
- [10] H. Tian, Y. Chen, and C. Chang, "Dynamic-hash-table based public auditing for secure cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 701–714, 2017.
- [11] G. Xu, Y. Bai, C. Yan, and Y. Yang, "Check algorithm of data integrity verification results in big data storage," *Journal of Computer Research and Development*, vol. 54, no. 11, pp. 2487–2496, 2017.
- [12] T. Wang, M. Z. A. Bhuiyan, G. Wang, L. Qi, J. Wu, and H. Thairer, "Preserving balance between privacy and data integrity in edge-assisted Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2679–2689, 2019.
- [13] D. W. Chadwick, W. Fan, G. D. D. Costantino et al., "A cloud-edge based data security architecture for sharing and analysing cyber threat information," *Future Generation Computer Systems*, vol. 102, pp. 710–722, 2020.
- [14] X. Li, S. Liu, F. Wu, and P. C. R. Joel, "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4755–4763, 2018.
- [15] T. Wang, Y. Mei, and W. Jia, "Edge-based differential privacy computing for sensor-cloud systems," *Journal of Parallel and Distributed Computing*, vol. 136, pp. 75–85, 2020.
- [16] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008, <https://bitcoin.org/bitcoin.pdf>.
- [17] Ethereum, "Ethereum whitepaper," 2020, <https://ethereum.org/en/whitepaper>.
- [18] Hyperledger, "An introduction to hyperledger," 2018, https://www.hyperledger.org/wpcontent/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf.
- [19] W. Tsai, L. Yu, and R. Wang, "Blockchain application development techniques," *Journal of Software*, vol. 28, no. 6, pp. 1474–1487, 2017.
- [20] A. Kosba, A. Miller, E. Shi, Z. Wen, and P. Charalampos, "Hawk:the blockchain model of cryptography and privacy-preserving smart contracts," in *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP)*, pp. 839–858, IEEE, San Jose, USA, May 2016.
- [21] IBM, "Research leading block chain use cases," <https://www.ibm.com/blockchain/use-cases/>.
- [22] Microsoft, "What are blockchain-enabled digital ecosystems," 2020, <https://azure.microsoft.com/en-us/resources/what-are-blockchain-enabled-digital-ecosystems/>.
- [23] Amazon, "Amazon managed blockchain," <https://amazonaws-china.com/cn/managed-blockchain/>.
- [24] Y. Tian, J. Yuan, and H. Song, "Efficient privacy-preserving authentication framework for edge-assisted Internet of Drones," *Journal of Information Security and Applications*, vol. 48, Article ID 102354, 2019.
- [25] T. Wang, P. Wang, S. Cai, Y. Ma, A. Liu, and M. Xie, "A unified trustworthy environment establishment based on edge computing in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6083–6091, 2019.
- [26] D. Yue, R. Li, and Y. Zhang, "Blockchain-based verification framework for data integrity in edge-cloud storage," *Journal of Parallel and Distributed Computing*, vol. 146, pp. 1–14, 2020.
- [27] E. Bonna and J. Shiguang, "DecChain: a decentralized security approach in Edge Computing based on Blockchain," *Future Generation Computer Systems*, vol. 113, pp. 363–379, 2020.
- [28] Z. Ma, X. Wang, D. K. Jain, H. Khan, H. Gao, and Z. Wang, "A blockchain-based trusted data management scheme in edge

- computing,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2013–2021, 2020.
- [29] J. Kang, R. Yu, X. Huang et al., “Blockchain for secure and efficient data sharing in vehicular edge computing and networks,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2019.
- [30] K. Gai, Y. Wu, and L. Zhu, “Differential privacy-based blockchain for industrial internet-of-things,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4156–4165, 2020.
- [31] B. Li, Q. He, F. Chen, H. Jin, X. Yang, and Y. Yang, “Auditing cache data integrity in the edge computing environment,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1210–1223, 2020.
- [32] W. Tong, B. Jiang, F. Xu, X. Lu, and Z. Sheng, “Privacy-preserving data integrity verification in mobile edge computing,” in *Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1007–1018, IEEE, Dallas, TX, USA, July 2019.
- [33] G.-X. Liu, L.-F. Shi, and D.-J. Xin, “Data integrity monitoring method of digital sensors for Internet-of-Things applications,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4575–4584, 2020.
- [34] G. S. Aujla, A. Singh, and M. Singh, “BloCkEd: blockchain-based secure data processing framework in edge envisioned V2X environment,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5850–5863, 2020.
- [35] J. Wu, *The Research of Fault Detection and Event Detection for Wireless Networks*, Nanjing University of Posts and Telecommunications, Nanjing, China, 2013.
- [36] X. Xu, P. Gong, Y. Zhang, and C. G. Bi, “Mobile-agent-based composite data destruction mechanism for cloud-P2P,” *Computer Science*, vol. 42, no. 10, pp. 138–146, 2015.
- [37] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the Weil pairing,” *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.
- [38] J. Benet, “Ipfs-content addressed, versioned, p2p file system,” *Eprint Arxiv*, vol. 07, no. 2, pp. 23–34, 2014.
- [39] S. Tan, Y. Jia, and W. Han, “Research and development of provable data integrity in cloud storage,” *Chinese Journal of Computers*, vol. 38, no. 1, pp. 164–177, 2015.
- [40] C. Liu, J. Chen, L. T. Yang et al., “Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2234–2244, 2014.
- [41] Z. Qin, S. Wu, and H. Xiong, “A review on data integrity protocols for data storage in cloud computing,” *Netinfo Security*, vol. 7, pp. 1–6, 2014.
- [42] S. S. H. H. J. Yau, “Dynamic audit services for outsourced storages in clouds,” *IEEE Transactions on Services Computing*, vol. 6, no. 2, pp. 227–238, 2013.

Research Article

BSD-Guard: A Collaborative Blockchain-Based Approach for Detection and Mitigation of SDN-Targeted DDoS Attacks

Shanqing Jiang ^{1,2,3}, **Lin Yang** ², **Xianming Gao** ², **Yuyang Zhou** ^{1,3,4}, **Tao Feng** ²,
Yanbo Song ⁵, **Kexian Liu** ⁶ and **Guang Cheng** ^{1,3,4}

¹School of Cyber Science and Engineering, Southeast University, Nanjing, China

²National Key Laboratory of Science and Technology on Information System Security, Institute of System Engineering, PLA Academy of Military Science, Beijing, China

³Purple Mountain Laboratories, Nanjing, China

⁴Jiangsu Province Engineering Research Center of Security for Ubiquitous Network, Nanjing, China

⁵State Key Laboratory on Integrated Services Networks, Xidian University, Xi'an, China

⁶School of Computing Science, Beijing University of Posts and Telecommunications, Beijing, China

Correspondence should be addressed to Guang Cheng; gcheng@njnet.edu.cn

Received 3 January 2022; Revised 16 February 2022; Accepted 24 February 2022; Published 12 April 2022

Academic Editor: Yuling Chen

Copyright © 2022 Shanqing Jiang 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.

Software-Defined Networking (SDN) enhances the flexibility and programmability of networks by separating control plane and data plane. The logically centralized control mechanism makes the control plane vulnerable in both single and multiple controller scenarios. Malicious third parties can exploit vulnerabilities of reactive forwarding mode to launch distributed denial-of-service (DDoS) attacks against SDN controllers. Unfortunately, existing DoS/DDoS solutions under single controller can not afford effective performance under multiple controllers due to the absence of cooperative detection and mitigation. To solve the above problem, we propose a blockchain-based SDN-targeted DDoS defense framework (BSD-Guard) that can provide cooperative detection and mitigation mechanism to protect SDN controllers. BSD-Guard introduces a blockchain-based secure middle plane between control plane and data plane. The secure middle plane calculates the suspect rate of new flows based on the collected packets' information and reports suspect lists to blockchain for immutably storing and sharing. Besides, the smart contract deployed on blockchain in advance constitutes collaborative defense strategies based on the suspect lists reported from multiple SDN domains. When receiving defense strategies, the secure middle plane converts them to specific flow table actions and installs actions into relevant switches. The experimental results indicate that BSD-Guard can efficiently detect DoS/DDoS attacks in multiple controllers scenario and issue precise defensive strategies near the source of attack by identifying the attack path.

1. Introduction

Software-Defined Network (SDN) is a novel network architecture designed to help network operators better manage infrastructures. The separation of control and data planes and logical centralized control bring network with high availability and programmability [1]. The logical centralized control plane conducts the behaviors of data plane via southbound protocols, in which the OpenFlow has developed as a typical and widely used southbound protocol. OpenFlow allows reactive mode for installing forwarding

rules, which has greatly simplified the rules configuration and policy deployment. The reactive mode arranges the table-missed packets to be encapsulated into *packet_in* message and reports to control plane for generating new forwarding rules. Because of the limited computational capacity, controller may discard normal requested traffics when the number of table-missed packets exceeds controller's processing capacity. This vulnerability can easily be exploited by attackers to launch resource-exhausting attacks against SDN controllers. Among them, the DoS/DDoS attack against SDN controllers has become a critical problem

[2] in recent years. Since controller determines the computation of end-to-end transmission path, when controllers suffer DoS/DDoS attack, it will disturb the normal message forwarding in the control domain, which may further cause the whole network to be disrupted. In summary, DoS/DDoS attack has become a serious security risk, affecting SDN architecture for the following reasons. First, the interaction mechanism based on OpenFlow protocol makes the controller a target of malicious attackers [3]. Second, the attacks are inexpensive and convenient to be implemented through launching forged requesting messages at hijacked hosts. Third, the attacker cannot be accurately traced and legitimate messages may be misdiscarded during defense process.

Traditional DDoS detecting and mitigating methods are often divided into four levels: attack detection, load balancing, traffic filtering, and traffic analysis [4]. (1) Attack detection is to identify DDoS traffic from normal traffic. Common detection methods are mainly based on message statistics and machine learning, which need to be ensured in real-time and accuracy of detection. (2) Load balancing relieves the storage and computing pressure of the victimized target by rerouting or traffic migrating and provides a brief resistance to sudden abnormal traffic within the tolerable range of the load balancing module. (3) Traffic filtering discards DDoS attack traffic by identifying the abnormal traffic characteristics, with the goal of improving the accuracy of identification and ensuring that normal traffic can be forwarded normally by network devices. (4) Traffic analysis aims to identify the attacker's intended behavior and trace the source of the attack by analyzing the collected attack traffic data.

There are two main shortcomings in current research on DoS/DDoS defense for SDN controllers. First, the misclassification of detection may cause the first packet of legitimate traffic to be discarded. Unlike the packet retransmission mechanism in traditional networks, in the SDN environment, the first packet being dropped will lead to more serious consequences. The subsequent messages of the normal traffic will not be processed, and the first packet request needs to be initiated again until controller issues correct forwarding rules. This blocking process affects the communication of normal service in data plane, increasing the burden of controllers and southbound channel. Second, the traditional DDoS detection and mitigation solutions for single controller can not be directly applied to multiple controllers scenario. In multiple controllers scenario, it is difficult to detect large-scale and distributed DDoS attacks and implement effective defense measures. Although the centralized control of SDN provides convenience for the monitoring of network status, for large-scale DDoS attacks, the centralized detection method is redundant and expensive, and the threat situation of the whole network has not been utilized reasonably. The east-west interface is used to maintain communication between controllers, which has not been standardized and not enough to support collaborative awareness and defensive decision. The collaborative protection mechanisms against DDoS in multicontroller scenario have also been introduced in recent years. In [5], a collaborative DDoS defense system can reroute crashing traffic to

other domains for filtering. In [6], the Redis Simple Message Queue (RSMQ) approach was used to collaboratively share detection and mitigation rules among multiple controllers. The latest research begins to seek cooperation with blockchain, and its decentralized features bring convenience to collaborative detection and mitigation. Researches [7–9] proposed a blockchain-based SDN framework to share threat information between multiple controllers. However, the smart contract was only used to share risky IP address and the time consumption of generating new block in Ethereum reaches flagrant 14 seconds [10].

In this paper, we propose BSD-Guard, a collaborative and elastic blockchain-based detection and defense system to protect SDN against controller targeted DDoS attacks. BSD-Guard stands between control plane and data plane, consisting of blockchain-based secure middle plane. In the detection stage, the secure middle plane collects statistics information about packets and ports from edge switches. Then the suspect lists calculated by detection algorithm are shared on the blockchain that can not be tampered by malicious attackers. And a global threat situation can be generated by cooperating smart contracts among multiple SDN domains. In the mitigation stage, the defensive strategies generated on blockchain can be installed into the edge switches by secure middle plane. Finally, the attacking packets can be discarded at source switches and benign packets can be forwarded correctly. And the SDN controllers can maintain a low level of CPU utilization when DDoS attack occurs. Our main technical contributions are as follows:

- (i) *Novel Framework*. We propose a novel detection and defense framework for protecting SDN controllers from DDoS attacks. The secure middle plane can perform as a proxy for a controller to detect and discard abnormal traffics. The blockchain becomes a platform for information sharing and defense policies scheduling between multiple controllers.
- (ii) *Fine-Grained Detection*. We present an entropy based suspect rate calculation method for fine-grained DDoS detection. The blacklist and graylist are generated by the type of forged addresses and its suspect rate. The fine-grained suspect list is beneficial for the subsequent development of precise defense strategies.
- (iii) *Collaborative Mitigation*. The detection and mitigation smart contracts deployed on the blockchain can collaborate with threat information reported by multiple secure middle planes. It can accurately identify the scale and the path of DDoS attacks and develop targeted defense strategies.

The rest of this paper will be organized as follows: Section 2 introduces the related works of existing detection and defense of DDoS attacks in the SDN environment. In Section 3, we present the problem statement about the adversary model and attack scenario. In Section 4, we introduce the detailed designs of the BSD-Guard system. Section 5 is the implementation and experimental evaluation of BSD-Guard. Finally, we make the summary of this paper in Section 6.

2. Related Works

2.1. Detection and Mitigation Methods under Single Controller. The DDoS attack on SDN controller has become a serious problem that can affect cloud environments and industrial production platforms that run over SDN networks, which will cause severe network security incidents. To solve this problem, researchers have proposed a large number of detection and defense solutions under single controller scenario. The DDoS detection solutions can be categorized into statistics-based schemes and machine learning based schemes under single controller.

Firstly, the statistics-based detection and mitigation scheme identifies DDoS attack by extracting statistical features of data plane traffic. In [11, 12], researchers identified DDoS attack traffic by detecting the rate and characteristic value of *packet_in* messages. You et al. [11] deployed the traffic collection module on the controller to collect, parse, and extract feature information of *packet_in* and calculate the rate of *packet_in*, entropy value of destination IP address, and port number. Huang et al. [12] predicted the number of *packet_in* in the next cycle by Taylor's formula; the detection module will be activated when the number exceeds the threshold. Then the characteristic values of *packet_in* were extracted for entropy calculation and determined whether there is a DDoS attack according to the entropy value. In [13–16], researchers have also advanced statistical analysis methods of flow tables to detect DDoS attacks. Fouladi et al. [13] detected DDoS by time series analysis of flow tables and determined the aggregation of traffic in network using feature information of destination IP address. Through the extraction of flow table features, the source of attack can also be traced back. Hassan et al. [16] used a lightweight approach to detect and defend against DDoS in SDN based on *Tsallis* entropy, which is able to detect DDoS at early stages, and the proposed dynamic threshold mechanism allows the detection method to adapt to dynamically changing network conditions. There also exist some studies that extract statistical feature from *sFlow* (Sampled Flow) to identify DDoS traffics [17–19]. Lawal et al. [17] obtained *CounterSample* and *FlowSample* messages by *sFlow* sampling, extracting traffic features, calculating traffic rate, and determining the presence of DDoS attack in real time by setting thresholds. Kumar et al. [18] obtained the feature values of traffic by *sFlow* and used machine learning for DDoS detection. Lu et al. [19] employed *sFlow* to obtain packets rate and aggregation of destination IP address in SDN network and jointly determined whether suspicious traffic occurred in SDN. In [20], Chen et al. proposed SDNShield, a three-stage overload control scheme for mitigating DDoS in SDN based on NFV technologies. The simulation results showed that SDNShield can achieve resilient performance against brute-force DDoS attacks and maintain excellent flow service quality at the same time.

Secondly, the machine learning based detection and mitigation scheme identifies and classifies DDoS traffics by various machine learning methods. Mehr and Ramamurthy [21] used the support vector machine to detect DDoS attacks

and install defense flow table entries to the switch, which reduced the impact of DDoS attacks on Ryu controllers by 36%. Considering the imbalance of traffic distribution, Cui et al. [22] introduced clustering algorithms such as the k-means to detect malicious traffics. In addition, the authors used *packet_in* message register to filter malicious traffic and evaluate the scheme in terms of detection accuracy, defense effectiveness, and communication latency. Some other researchers proposed hybrid machine learning approaches. Deepa et al. [23] proposed a model of hybrid machine learning with support vector machines and self-organizing mappings, which can effectively protect the SDN controllers to work properly when DDoS attacks occur. Nugraha and Murthy [24] proposed a hybrid Convolutional Neural Network-Long-Short Term Memory (CNN-LSTM) model to detect slow DDoS attacks in SDN networks, and experiments showed that the method achieved 99% accuracy in the considered performance metrics. Xu et al. [25] proposed an efficient and accurate DDoS detection method based on SDN cloud edge collaboration. The method used an entropy approach to select ideal SOM mappings and classify SOM neurons, and then KD-trees were used to identify traffics at a finer granularity, which improved the accuracy of DDoS detection. Ujjan et al. [26] proposed a DDoS detection method based on adaptive polling sampling of *sFlow* and deep learning models. Adaptive polling sampling of *sFlow* was used in the data plane to reduce the switch's overhead. Snort IDS and SAE deep learning models were deployed in the control plane to improve the accuracy of detection. The authors quantitatively investigated the trade-off between the accuracy of attack detection and resource overhead. Luong et al. [27] proposed a DDoS detection model in SDN based on machine learning and deep neural networks, and authors compared the model with decision trees and random forest models. The results showed that complex DDoS detection systems do not necessarily produce more accurate results than simple ones.

2.2. Cooperative Defense under Multicontrollers. DDoS attacks are complex and varied in the actual network environment. Attackers often launch DDoS traffics from remote locations to one target by hijacking a large number of puppet hosts or exploiting vulnerabilities in existing communication protocols. The DDoS attack under multiple controllers network has become one of the most difficult threats in SDN environment. This is because traffics within disparate controller domains often exhibit different characteristics, and a more concentrated aggregation of abnormal traffic usually emerges in the victim's domain. The domain is defined as the partial network managed by one controller. Usually, the DDoS attack has already been carried out in the source domain when detection mechanism was triggered in the targeted domain, which will leave the defender quite limited time to respond and defense. Therefore, it is necessary to share threat information among multiple domains to identify and intercept abnormal traffic during the initiation and dissemination phase, which will save more time for protecting the target controller.

There exist several studies on cooperative defending against DDoS attacks. The IETF is proposing an ongoing protocol called DOTS (DDoS Open Threat Signal) [28], which will mitigate DDoS attacks by an intradomain and cross-domain collaborative solution. The servers and clients of DOTS are required to broadcast blacklists or whitelists addresses. When detecting attacks, the client requests mitigation services from the server responsible for cross-domain communication and coordination. However, the DOTS is still faced with implementation complexity to support different types of communication in distributed and centralized architectures. A similar approach is presented in [29]. The authors employ an advertising protocol based on FLEX (Flow-based Event eXchange) format to simplify the deployment and collaboration between domains. This protocol supports realizing the situational awareness of the current threat posture, pooling expertise and resources, and facilitating automated defense against persistent cyberattacks. However, the deployments of above solutions are complex since they need to create or modify protocols for distributed network architectures. Instead, these collaboration requirements can be met by the natural characteristics of SDN, blockchain, and smart contract, thus avoiding the complexity of deployment and adoption of new protocols.

Blockchain and smart contracts have shown their unique advantages in the area of collaborative threat detection and defense for SDN and IoT. Javaid et al. [30] introduced a smart contracts-enabled IoT device communication framework using Ethereum, a blockchain variant to replace the traditional centralized IoT infrastructure. Smart contracts are required for IoT devices accessing the network. And trusted or untrusted devices can be distinguished by the proposed system. Shao et al. [31] proposed a blockchain-based SDN security system model and a consensus algorithm SPBFT to improve the security and consensus efficiency of the SDN control plane. The smart contracts periodically check the status of controller to detect DDoS attack. Abou et al. [7, 8] designed a collaborative distributed DDoS mitigation framework based on blockchain. The framework utilized smart contracts to transfer attack information between SDN multiple domains to reduce the huge cost of forwarding useless packets across multiple domains. Extensive experiments on both private and public networks (Ganache simulator, Ropsten test network) show that Cochain-SC achieves versatility, security, efficiency, and cost-effectiveness. In [9], a blockchain-based SDN architecture was proposed to advertise whitelisted or blacklisted IP addresses to defend against DDoS attacks, enabling the execution of defense rules across multiple domains. However, the advantages of blockchain and smart contracts have not been fully exploited in existing research. As an excellent distributed collaborative platform, blockchain should not be limited to sharing blacklisted and whitelisted IP addresses, but also sharing the data plane traffic characteristics that are originally opaque between multiple controllers. This method will allow the characteristics of DDoS attack to be

jointly discovered at an earlier stage. And smart contracts can also be used as triggers for issuing defense policies automatically.

3. Problem Statement

In this section, we first introduce the workflow of handling normal traffic in SDN networks. Then we present the adversary model of SDN-targeted DoS and DDoS attacks. Finally, we state the challenges of detecting and mitigating the DDoS attacks in multiple SDN controllers networks and the basic principles that should be kept in the process.

3.1. SDN Workflow. OpenFlow has become a widely used standard southbound interface protocol that specifies the pipeline for switches to handle packets and the types of messages between the data plane and the control plane. OpenFlow supports both proactive and reactive approaches to install flow forwarding rules. In the proactive mode, the controller preregisters forwarding rules on switches to handle incoming packets. In the reactive mode, when an OpenFlow switch receives several new incoming packets, it will process each packet by following steps with the FIFO (first input first output) manner [32], as shown in Figure 1.

- (1) The OpenFlow Agent (OFA) traverses its flow table to find if there exist flow table entries that match the header of the new-coming packet. If a match occurs, the switch will process the packet according to the action field of the flow table entry, such as forwarding. Otherwise, the switch treats the packet as *table-miss* by caching it into the buffer area, encapsulating its header into *packet_in* message, and sending to the controller. If the buffer is full, the entire packet will be encapsulated into a *packet_in* message (Steps A, B, and C).
- (2) The SDN controller receives the *packet_in* message and calculates forwarding policy based on the global networking view and applications' intention. The forwarding action will be encapsulated into a *packet_out* message and sent back to the switch (Steps D, E).
- (3) The OFA receives the *packet_out* message and installs the entries into the flow table and then handles the buffered table-miss packet based on the instruction in *packet_out* (Steps F, G).
- (4) When the further packets with the same header arrive within the survival time of flow table entries, the OFA can deal with these packets according to the "match" and "action" instruction with linear rate.

This reactive flow table installation method enables a flexible way to control network traffic, which is the core principle of SDN's control and forwarding decoupling. It is widely used in most OpenFlow scenarios. However, due to the limited processing capacity of hardware and software, this method has also become a source of resource-consuming threats in sSDN networks.

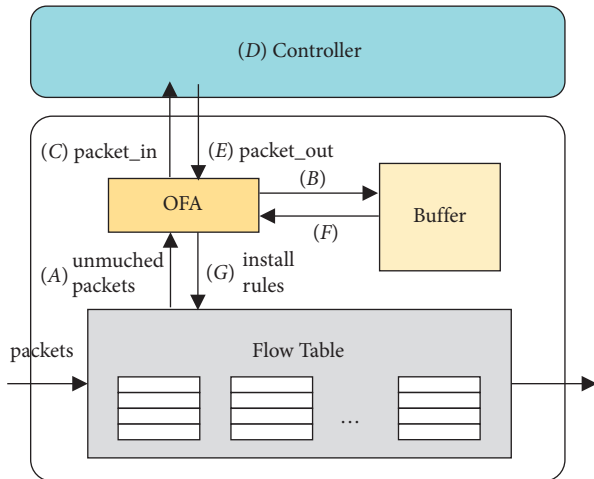


FIGURE 1: The workflow of OpenFlow reactive method.

3.2. Adversary Model. The SDN-targeted DoS/DDoS attacks are different from DoS/DDoS attacks in traditional networks. The reactive method of SDN's workflow can be exploited by tricky attackers to launch DoS/DDoS attacks. When encountering table-missing packets, the OpenFlow switch must initiate a *packet_in* message to controller for requiring forwarding actions. By sending a large number of forged address packets, attackers can stimulate switch with abundant meaningless *packet_in* messages to controller, which will result in excessive consumption of CPU and storage resources, meanwhile causing switch's buffer overflowing and control channel blocking. When an attacker injects a large number of forged new packets into multiple switches in the data plane at the same time, the controller will suffer a more serious DDoS attack, as shown in Figure 2.

We demonstrate the damage of SDN-targeted DDoS attacks with a group of experiments. We set up an experimental environment consisting of one ONOS controller and 10 OpenVSwitch in mininet (with linear topology). In the first DoS group, forged address packets are injected through one switch with 20 pps to 200 pps. In the second DDoS group, forged address packets are injected through 10 switches with 20 pps to 200 pps concurrently. The result in Figure 3 shows that the CPU utilization rate of the ONOS controller in the DDoS group increases higher than DoS. From the attacker's view, the DoS attack with 200 pps and DDoS attack with 20 pps * 10 will stimulate the same number of new *packet_in* message theoretically. However, we can find in Figure 3 that when the total forged packets rate is 200 pps, the CPU utilization rate in the DDoS group (attack intensity = 20 pps * 10, CPU = 44.41%) is much higher than that in the DoS group (attack intensity = 200 pps * 1, CPU = 11.6%). Therefore, we can conclude that a DDoS attack launched from multiple switches has more serious harm to the control plane than a DoS attack when attackers equip limited attack resources. Besides, we also make another interesting comparison. We disconnect the links between 10 switches and perform DDoS attack again, and the result shows that the CPU consumption of controller decreases by 20% averagely. By capturing packets and analyzing, the truth is that when switches are linked with

each other, the forged new packets of DDoS can be broadcasted among switches, which makes the number of *packet_in* messages reported to controller be amplified. Above all, the control plane will be more vulnerable to DDoS attacks under a distributed network scenario.

3.3. Scenario and Challenges. The multiple SDN controllers environment is displayed in Figure 4. In the multiple controllers' scenario, the controller targeted DDoS attacks could be launched from remote data plane managed by other controllers. To implement DDoS attacks more stealthily, the tricky attackers often initiate attacking packets from the neighboring domains of the victim controller, which greatly increases the difficulty of detection and mitigation. And the defensive actions performed in the victim's domain will not be effective to mitigate such attacks. There have been a lot of previous researches on how to detect DDoS attacks between multiple controllers [7, 33]. However, the fine-grained threat information can not be shared collaboratively across multiple controllers, and a complete set of defense schemes has not been developed.

Although there has been a lot of valuable researches on DDoS detection and defense for protecting controllers, two key issues remain unresolved. First, existing detection processes require data plane traffic and network state information to be reported to the centralized controller, which will greatly increase the burden of controller and south-bound channel. Second, in the multiple controllers' scenario, threat information within a domain can only be mastered by the internal controller. However, the threat state perceived in one domain is only local information, which cannot form the most effective defense plan. Meanwhile, the interaction among east-west interface will consume the resources for synchronizing state information. Although the author in [7] proposed a blockchain-based framework Cochain-SC to facilitate the collaboration for smart contract-based intradomain DDoS mitigation, the sharing information between intradomains is limited to blacklisted IPs.

Therefore, in the multiple controllers' DDoS defense scenario, collaborative integration of threat information and network resources between north-south and east-west needs to be considered simultaneously. From the north-south view, the threat situation of DDoS traffic in the data plane should not be completely reported to the control plane, which can reserve the valuable computing resources of controller and avoid single-point failure. From the east-west view, the more fine-grained network threat information from multiple control domains should be shared for identifying attack scenarios and tracing attacker more precisely and preventing DDoS traffic from spreading among multiple controllers.

4. System Overview

We design a system named BSD-Guard, which can detect and mitigate SDN-targeted DDoS attacks among multiple controllers. This system can calculate suspect lists according to traffic statistical information from multiple controllers. The

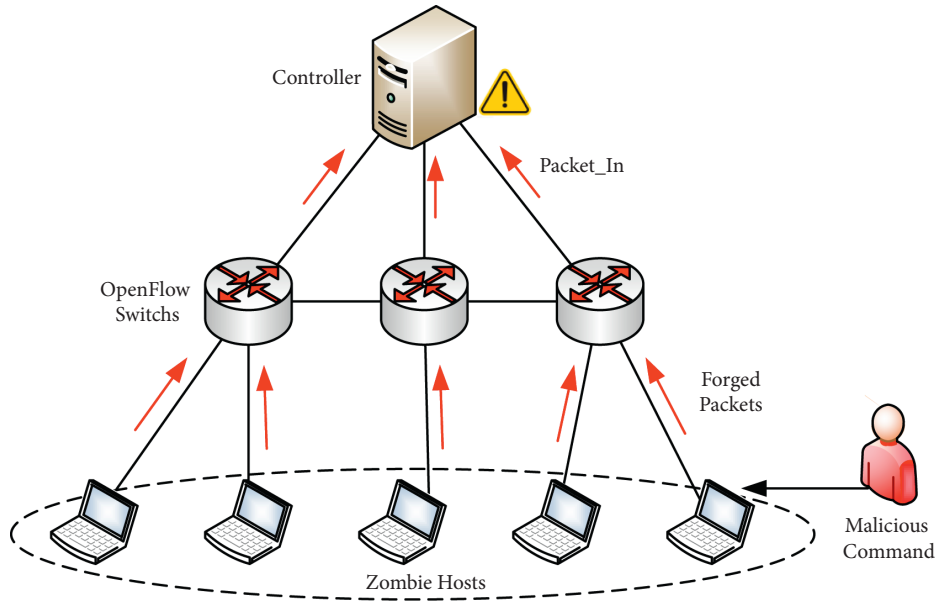


FIGURE 2: The controller targeted DDoS attack in SDN.

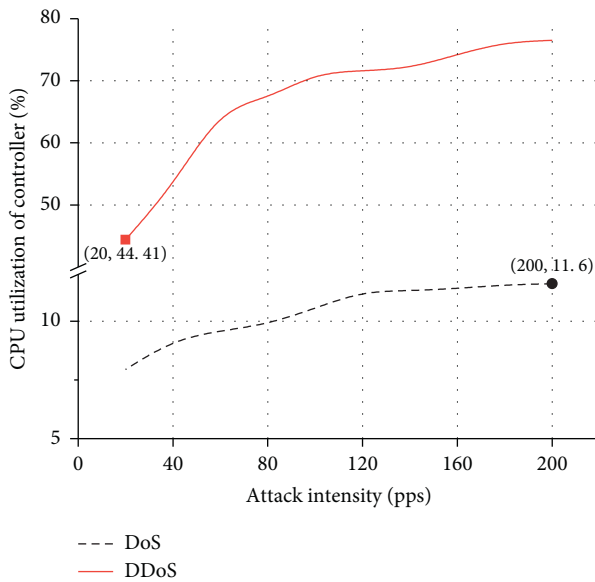


FIGURE 3: The CPU utilization rate of ONOS controller under DoS and DDoS attacks (the upper limit of the Y-axis in DoS group is 12% and is 80% in DDoS group).

threat information can be shared on blockchain via smart contracts. The detection module can collaboratively detect DDoS attacks among multiple controllers and trace the source of attack. The mitigation module can issue defense policies near the source of attack. And the defense strength can be adjusted in conjunction with the controller's real-time load to reduce the misdiscarded rate of normal traffics.

4.1. Architectural Components Overview. BSD-Guard consists of two main modules: secure middle plane and blockchain, as displayed in Figure 5. The secure middle plane

stands between the control plane and the data plane, which contains threat detecting and policy issuing functions and smart contract APIs interacting with the blockchain. The threat detecting function collects *sFlow* and intercepts *packet_in* messages sent from data plane to control plane. The policy issuing function receives DDoS mitigating policies from blockchain and registers flow rules into the intrusive switches. The smart contract APIs are responsible for reporting threat information to the blockchain and receiving cocculated defense strategies. The blockchain plays the role of storage and collaborative sharing of threat state information for multiple SDN domains. It contains blockchain nodes and smart contracts. The DDoS threat information of multiple controllers can be aggregated in blockchain to identify the cross-domain DDoS attack behavior. And the information stored on blockchain can not be tampered by malicious attackers.

In terms of workflow, the system is divided into detection stage, collaboration stage, and mitigation stage. The complete processes are introduced as the following seven steps, and the interaction flow is shown as Figure 6.

- (1) In each SDN domain, the threat collecting module collects *sFlow* countersample messages periodically by *sFlow* agents deployed on each OpenFlow switch. Once detecting the velocity of flows exceeds the specified threshold, the detection program records the corresponding switch's *IP* and *port*.
- (2) The secure middle plane resolves the *packet_in* messages collected from OpenFlow switches whose port is overspeeding. The *data* field will be extracted for inspecting the original message that triggers table-missing on switch.
- (3) The fields extracted from *packet_in* are used to periodically calculate the suspect rate of new flow. And the black/graylists (including *SwitchIP*, *Port*, *IP*, *Mac*,

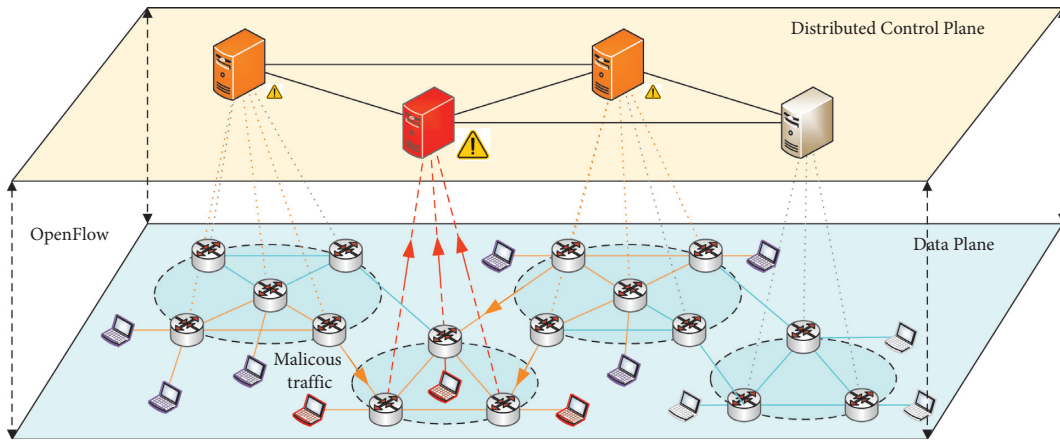


FIGURE 4: The DDoS scenario in SDN with multiple controllers.

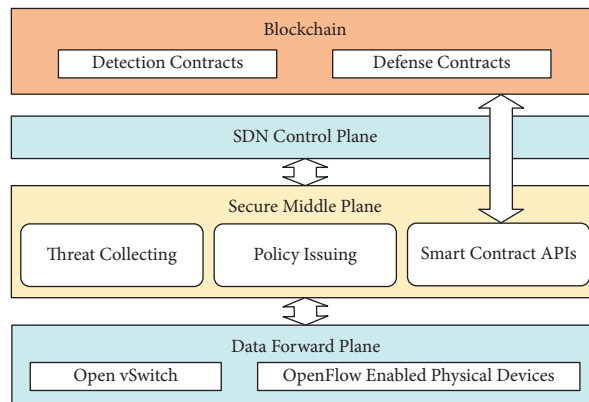


FIGURE 5: The overview architecture of BSD-Guard.

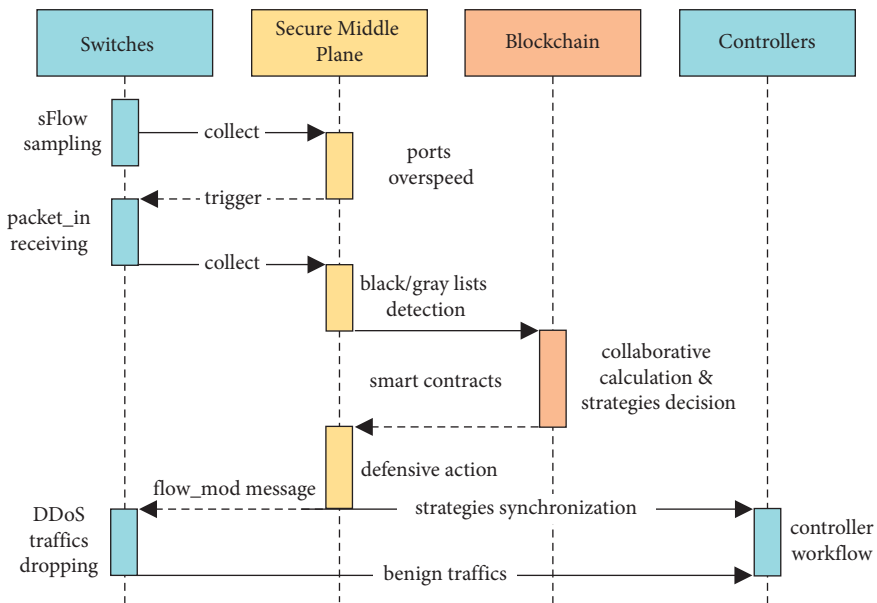


FIGURE 6: The interaction flows of BSD-Guard.

and suspect rate) will be reported upon blockchain through the smart contract.

- (4) The data plane network topologies of multiple controllers are also recorded on the blockchain. Once changed, the real-time topology will be updated by the smart contract.
- (5) The collaborative detection algorithm combines the suspect lists and multiple controllers' topology in the previous steps and identifies a complete attack path on the blockchain.
- (6) The mitigation algorithm establishes the defense strategies according to the detection results and issues the corresponding instructions to the secure middle plane of the victim controller.
- (7) Each secure middle plane executes the defense actions based on the received strategies and controller's real-time load elastically. The *flow_remove* and *flow_mod* message are installed into switches and synchronized to the controller to clear meaningless flow table entries and issue new defense flow table entries.

4.2. Fine-Grained Detection Based on Suspect Rate. Based on the analysis of adversary model, the controller targeted DDoS attack is launched from data plane. Large amounts of forged packets trigger switches sending many meaningless *packet_in* to the controller. In order to detect DDoS attack launched from data plane in a timely manner, we choose the sFlow protocol, an efficient and flexible approach that does not consume the computing capacity and bandwidth of SDN controller and OpenFlow switches. The sFlow agent deployed on switches can generate *FlowSample* and *CounterSample* messages and send them to the sFlow collector with a fixed period. We can calculate the packet rate of switches' inport and outport by analyzing *CounterSample* messages. When the inport packet rate of a switch is detected to exceed the normal threshold, it is considered that a DDoS attack may occur. Then the *packet_in* parsing module is activated to extract the original packets that crash from the overspeed port. These information will be recorded into Elasticsearch (ES) database with six elements' tuple: $\langle \text{SwitchID}, \text{InPort}, \text{SrcMac}, \text{SrcIP}, \text{DstMac}, \text{DstIP} \rangle$. These tuples will be counted with several attributes in each fixed period, which is the same as the sampling period of *CounterSample*. These attributes describe the forged level of *Mac* and *IP*, as the SDN-targeted DDoS attack is mainly launched by forged packets that do not match the existing flow table on the OpenFlow switch. The meanings of these attributes are listed in Table 1. The suspect lists and suspect rate will be calculated based on these attributes in the following description.

Inspired by [20], we present an entropy based calculation method of packet's suspect rate. For each element in Table 1, we use the frequency of each element to approximately estimate its probability in each statistical period.

$$p_i = \frac{X_i}{\sum_{i=1}^N X_i}. \quad (1)$$

TABLE 1: The characteristics of six elements' tuple in each period.

Abbreviation	Explanation
<i>Switch_IP</i>	The IP address of switch
<i>Inport</i>	The overspeed port on switch
<i>SrcMac_num</i>	The arising number of <i>SrcMac</i>
<i>SrcIP_num</i>	The arising number of <i>SrcIP</i>
<i>DstMac_num</i>	The arising number of <i>SrcMac</i>
<i>DstIP_num</i>	The arising number of <i>DstIP</i>

X represents the element in each tuple (X will be replaced by *SwitchID*, *InPort*, *SrcMac*, *SrcIP*, *DstMac*, *DstIP* in the calculation process), and the lower corner i represents the item number of the *packet_in* during the statistical period. Then we can get the information entropy value of each attribute in the tuple by

$$H(X) = - \sum_{i=1}^N p_i \log p_i. \quad (2)$$

The information entropy in (2) has been widely used in detecting DDoS traffics. It shows good performance in demonstrating the discrete degree of statistical features. However, it can only represent the overall dispersion of an attribute during the statistical period and cannot pinpoint which specific item affects the entropy value. Therefore, we introduce entropy based suspect rate calculation method, which combines the entropy of each attribute with the frequency of occurrence of the corresponding item. f is a new flow, representing a *packet_in* item in ES database. f_i represents one of the attributes in six elements' tuple. For example, $H(f_{srcmac})/p(f_{srcmac})$ reflects the suspicious level of *SrcMac* in this *packet_in* f . If the *SrcMac* is forged by random generating, $H(f_{srcmac})$ is higher than normal level and the $p(f_{srcmac})$ value is less than normal level. Therefore, the calculated suspect_rate_f will be large when address forged DDoS attacks occur. The normal level of $H(f_j)/p(f_j)$ can be obtained in normal traffic scenario, recorded as θ_{normal} . For the attribute of *SrcMac*, *SrcIP*, *DstMac*, *DstIP*, if the value of $H(f_j)/p(f_j)$ is greater than θ_{normal} , the corresponding attribute can be judged as randomly forged. The malicious or hijacked host intends to stimulate the switch to generate a large number of *packet_in* to send to controller for consuming its computational load and storage. Therefore, we can detect the *packet_in* categories with different combinations of forged addresses. The real-time suspect lists can be figured out by periodically accessing the ES database updated in real-time. For the types of *packet_in* whose partial addresses are forged, the real address can be recognized into the blacklist and the suspect rate can be calculated based on (3). For the type of *packet_in* whose addresses are all forged, since the real source or destination address cannot be identified, only the corresponding overspeed switch's port can be recorded into graylist. Therefore, the graylist contains victim controller ID, switch IP and port, and suspect rate. The examples of graylist and blacklist are listed in Table 2.

TABLE 2: The example of graylist and blacklist stored on blockchain.

Type	SwitchIP	SwitchID	Port	DstIP	DstMAC	SuspectRate
GrayList	192.168.188.121	1c48cc37ab254bc1	ge-1/1/19	Null	Null	0.764 3
BlackList	192.168.188.199	45ac29bc3714dbc1	ge-1/1/3	192.168.188.201	9A-26-F7-08-0B-2 F	0.866 1

$$\text{suspect_rate}_f = \sum_{j \in \text{tuples}} \frac{H(f_j)}{P(f_j)}. \quad (3)$$

4.3. Suspect Lists Sharing Based on Smart Contract. Several detection and mitigation schemes have considered collaboration among multiple switches and controllers to tackle widely launched DDoS attacks. However, the complexity of deployment and the limitation of information sharing restrict the practical effectiveness of collaborative defense. Abou El Houda et al. [8] reported the suspect IP addresses from the victim domain to the collaborative domains by means of smart contract, which can block the illegal traffics in the source and intermediate domains. However, this approach can only deal with traditional DDoS attack against hosts. The sharing information only includes suspect IP addresses that cannot cope with more complex attack scenarios in which the IP addresses of malicious packets are forged. Considering this situation, we focus on SDN-targeted DDoS attacks in multiple controllers scenario and share the fine-grained DDoS threat information between multiple SDN domains. More specifically, through the collaborative sharing of the suspicious source or destination addresses (IP or Mac), the edge switch and port, and the suspect rate, a more precise detection and mitigation mechanism can be established.

We design two detection strategies in this collaboratively sharing mechanism with smart contracts. In the first strategy, we focus on the intradomain DDoS attack. Under the intradomain scenario, the destination Mac and IP of attack packets are randomly forged, which results in the forged packets not being forwarded to the neighboring domain. Therefore, only the intracontroller can suffer from a large number of meaningless *packet_in* request messages. In this case, if the source Mac or IP in the original packets is genuine, the corresponding packets will be precisely dropped at the edge switch by the blacklist strategy mentioned in Section 4.2. However, if the tricky attacker forges all the source Mac and IP, it is impossible to locate the specific puppet host, and only the abnormal switch's port can be determined. This situation makes the defender very embarrassed. If discarding all messages coming from that switch's port, the normal service traffic will be affected innocently. And if multiple switches are injected with low-intensity forged packets, it will escape the threshold of single-point detection. To solve this problem, we deploy the smart contract to query the graylist related to the same controller on blockchain during the period. Multiple suspect lists from multiple switches are jointly calculated to derive the DDoS attack strength under the global view. This method avoids the failure of missing forged packets below the overspeed threshold on an individual switch.

In the second strategy, we focus on the controller targeted DDoS attack across domains. In the cross-domain scenario, the attacker can construct a large number of packets with forged source IP or Mac and real destination IP and Mac, which will stimulate the generation of *packet_in* of all switches on the path from attack source to destination. The controller issues forwarding rules based on the real destination address, so that the forged packets can be forwarded to the destination host hop by hop. Finally, the last-hop switch will be forced to generate abundant *packet_in* messages due to aggregation effect. As shown in Figure 4, the controller in destination domain suffers a serious DDoS attack from the neighbor domains. We collect *packet_in* messages of each overspeed switch port to calculate the blacklist (contains the controller ID, six elements' tuple, and suspect rate) and then store them on blockchain via smart contract. At the same time, the global topology and cross-domain links collated from each controller will also be uploaded to blockchain in real time. Once multiple blacklists with the same destination Mac or IP exist on the blockchain and the link formed by the suspect switches' ports conforms to the global topology, the link can be confirmed as the attack path of cross-domain DDoS. And the first-hop switch is the edge switch that brings in DDoS attack traffics. The cooperative detection algorithm of cross-domain DDoS attack is shown in Algorithm 1. The smart contract of blacklist is shown in Table 3, and the functions of smart contract consist of storing, searching, updating, and deleting blacklist. Once deployed on the blockchain, these functions can be executed automatically to share the blacklist of multiple control domain on the blockchain. Similarly, the smart contract of graylist equips the same functions to operate graylist on the blockchain.

4.4. Elastic DDoS Mitigation Based on Controller Load. We also design an elastic DDoS mitigation mechanism for different attack scenarios. In terms of the graylist scenario, a large number of meaningless *packet_in* come from switches within the controller domain and no valid blacklist features can be extracted from the raw data of messages. We develop a defense strategy to install *flow_mod* message to disable the graylisted switches' port that generated forged packets. And the disable time depends on the suspect rate and the real-time load of controller. Specifically, the field of *hard_time* in the *flow_mod* message can be calculated as

$$\text{hard_time} = SR_{ij} * \frac{\text{NumGL}_k}{\text{NumPktIn}_k} * 30(s). \quad (4)$$

SR_{ij} represents the suspect rate of port j on switch i , NumGL_k represents the number of graylists of controller k , and NumPktIn_k represents the total received *packet_in* of controller k during the continuous period. The benchmark

Input: *Graylists* on blockchain
Output: *flow_mod* message

- (1) Group the graylist by victim controller ID
- (2) **for** Each victim controller C_k **do**
- (3) Calculate the $NumGL_k$ and $NumPktIn_k$
- (4) **for** Each suspect switch **do**
- (5) Calculate the *hard_time* of each suspect port
- (6) Construct the *flow_mod* message according to *graylist* and *hard_time*
- (7) Issue the *flow_mod* message to switches
- (8) **end for**
- (9) **end for**

ALGORITHM 2: Elastic mitigation based on graylist.

module, DDoS detection module, defense policy issuing module, and smart contract APIs. All of them are deployed in *Docker* containers, which are convenient for management and migration. Meanwhile, we install the ONOS controller on the Huawei 2288H V5 server equipped with Intel(R) Xeon(R) Gold 6130 CPU and 64 GB memory. In terms of forwarding devices, we employ commercial OpenFlow switch Pica8 AS4610-54T to establish the data plane. We develop and install an application called *midonos* on ONOS to keep the communication between controller and secure middle plane. The blockchain is deployed among the secure middle planes with distributed blockchain nodes. We also use the Elasticsearch (ES) database to store *sFlow* and *packet_in* messages in a high-performance server. We employ four Ubuntu hosts as attacker, victim, and normal users, respectively, in our environment. The experimental network topology is displayed in Figure 7. There is no east-west interface between two ONOS controllers, and switches in the data plane have cross-domain links. We employ FISCO and WeBASE platform [34] to provide blockchain service. WeBASE is a set of common components built between blockchain applications and FISCO-BCOS nodes. It standardizes blockchain application development into five steps: deployment, configuration, development of smart contracts, development of application layer, and online management, which simplifies the process of deploying smart contracts. The interface of the node console (v2.8.0) is shown in Figure 8, which includes the management of blockchain nodes and smart contracts. Administrators can directly edit the contract's ".sol" file and then compile and deploy them on the blockchain.

5.2. Experimental Setup. We construct four experiments under two different attack scenarios to verify that our proposed system can detect and mitigate DDoS attacks. The effectiveness of the proposed system will be evaluated compared with the OpenFlow process without defense measures. First, we construct the DDoS attack scenario in one control domain. The $host_1$ and $host_2$ are selected as attackers to launch UDP flooding packets. We design two different address random methods of forged packets to verify the defense strategies specifically. The "UDP0000" represents the UDP flooding packets with randomly forged

$\langle SrcMac, SrcIP, DstMac, DstIP \rangle$. The "UDP1100" represents the UDP flooding packets with randomly forged $\langle DstMac, DstIP \rangle$ and genuine $\langle SrcMac, SrcIP \rangle$. The *Scapy* will be used to generate flooding UDP packets under 250 pps attack rate on two hosts. We measure the CPU utilization rate of ONOS controller and the rate of *packet_in* messages the controller received. Second, based on the previous scenario, we adjust the attack intensity of "UDP0000," ranging from 100 pps to 1000 pps. We also compare the CPU utilization of ONOS controller under the OpenFlow process and our BSD-Guard process. Third, we construct the DDoS attack scenario across two control domains. In Figure 7, the two controllers have no east-west interface, and the threat information of the two control domains is shared and synchronized through blockchain by the smart contract APIs on secure middle plane. We launch the TCP SYN flooding attack on $host_1$ with forged packets "TCP0011," which consists of randomly forged $\langle SrcMac, SrcIP \rangle$ and genuine $\langle DstMac, DstIP \rangle$ of $host_3$. Finally, we set up a comparative experiment to verify whether the proposed defense method interferes with normal traffic. We set the $host_2$ as a normal user and test whether the $host_2$ and $host_3$ can keep communication normally.

5.3. Experimental Result. In the first experiment, we compare our proposed system BSD-Guard with OpenFlow (no defense) mechanism under two types of UDP DDoS flooding (UDP0000 and UDP1100). The CPU utilization and received *packet_in* rate of ONOS controller are shown in Figure 9. In Figure 9(a), we launch UDP0000 flooding attack from two seconds to the end. It can be clearly seen that the *packet_in* rate and CPU utilization keep a continuously high level after the attack under OpenFlow scenario. Differently, under the BSD-Guard scenario, these two metrics rapidly decrease after five seconds because the defending *flow_mod* has been installed on the switch at the peak position of the curve. It can be validated by entering "ovs-ofctl dump-flows bridge" command on Pica8 switch, and the defending flow table entry contains suspicious inport generated by graylist. A similar phenomenon can be observed in Figure 9(b); the BSD-Guard takes only four seconds to detect and mitigate UDP1100 flooding attack. The response performance is better than the 13 seconds of Cochain-SC [7]. The *packet_in*

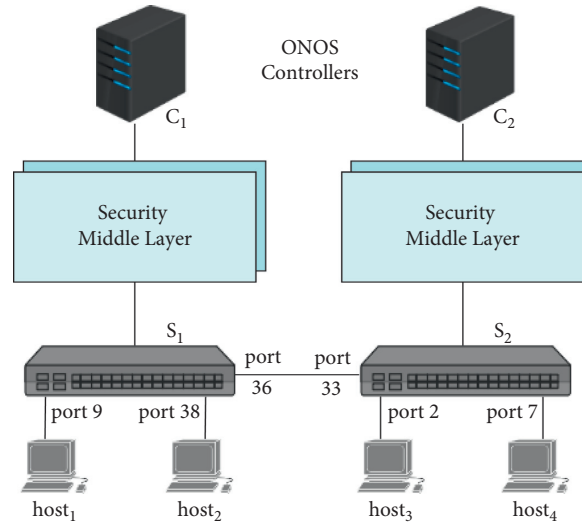


FIGURE 7: The experimental network topology.

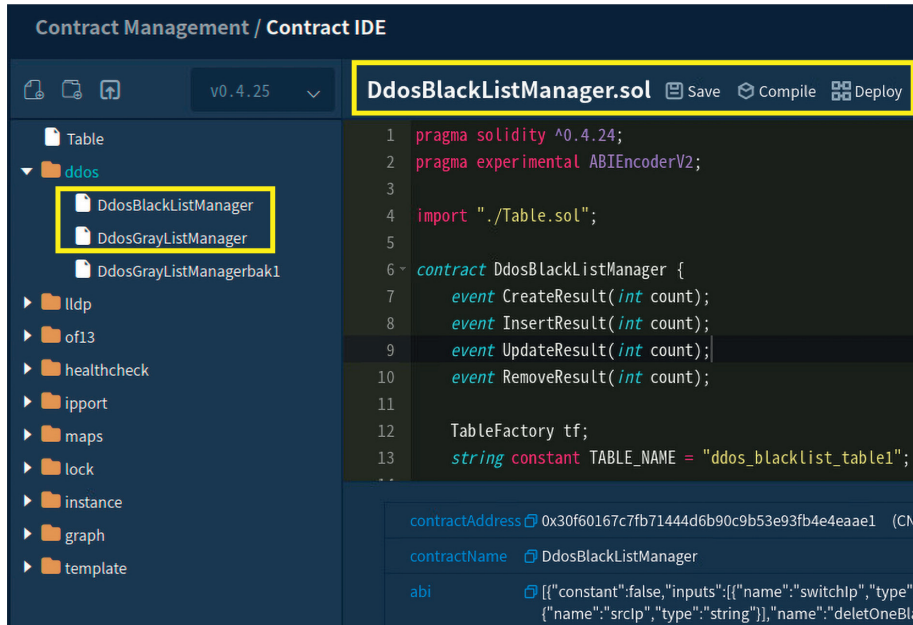


FIGURE 8: The management interface of smart contracts on WeBASE platform.

rate decreases from 779pps to 4pps within two seconds, and the CPU utilization of controller decreases from 22.5% to 5% within 2.5 seconds. Meanwhile, we can check the flow table on switch, which contains the entry with suspicious inport and $\langle SrcMac, SrcIP \rangle$ generated by blacklist, and the forwarding action is *drop*. It is worth noting that the CPU utilization of UDP0000 in Figure 9(a) is higher than UDP1100 in Figure 9(b) under OpenFlow mechanism with the same attack intensity. The randomly forged *SrcMACs* will be regarded as large amounts of new hosts in the controller's view. The creation and maintenance of new hosts' identity will consume a lot of CPU on the controller. In contrast, just forging the destination address will only make the controller consume CPU for calculating

forwarding rules, without creating new forged hosts. We also construct other types of packet forged methods, including UDP/TCP-1000/0100/1101. The results verify that our proposed system can report corresponding blacklists and install the special defense flow table according to the forged packet characteristics. This precise defense pattern can effectively avoid the incorrect discarding of normal packets.

In the second experiment, we adjust the intensity of DDoS attacks from 100pps to 1000pps and record the average CPU utilization of ONOS controller under OpenFlow and BSD-Guard mechanisms. As is shown in Figure 10, with the increasing of attack intensity, the CPU utilization of controller keeps rapid growth under OpenFlow scenario. The CPU utilization reaches amazing 85.6% when the attack

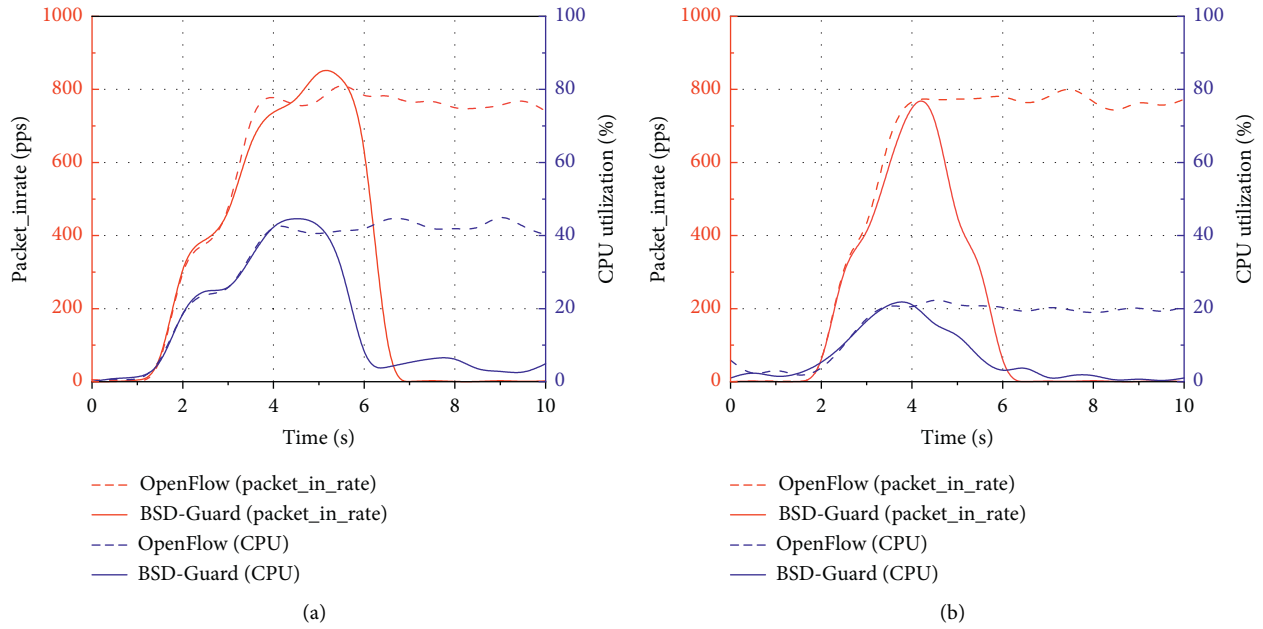


FIGURE 9: The *packet_in* rate and controller CPU utilization under udp0000 and udp1100 DDoS attack scenarios. (a) udp0000 attack; (b) udp1100 attack.

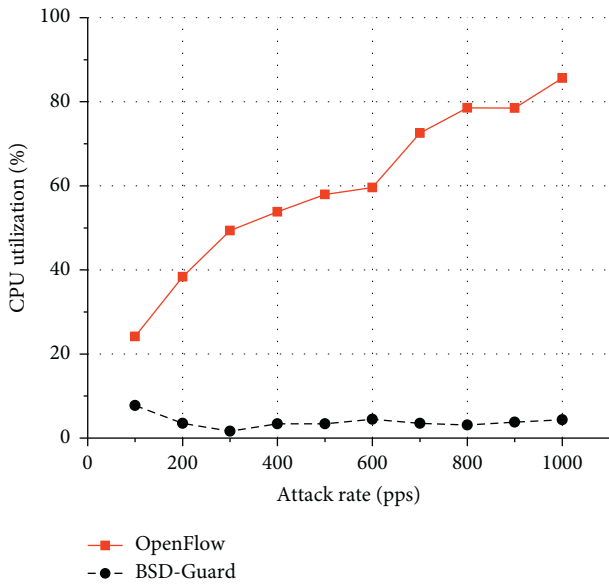


FIGURE 10: The CPU utilization of controller under different attack intensities.

rate is 1000 pps. In contrast, the values under BSD-Guard maintain a stable low level around 5%, which indicates that our proposed defense mechanism can resist high-intensity SDN-targeted DDoS attacks well.

In the third experiment, we construct a TCP SYN attack between two neighboring domains under the management of two ONOS controllers. The forged packet TCP0011 has randomly forged source addresses and genuine destination address of *host₃*, which allows the forged new packet to be forwarded to the target controller’s domain and reach the target *host₃*. In this process, the two controllers are

successively involved in the calculation of forwarding policies, as shown in Figures 11(a) and 11(b). The received *packet_in* rate and CPU utilization of the controller in source domain are earlier than the controller in targeted domain. The time difference between them is equal to the sum of switch forwarding delay and the delay caused by the source controller to make forwarding policies and install them to switch. Meanwhile, an attack path “*host₁* → *S₁* (*port₉*) → *S₁* (*port₃₆*) → *S₂* (*port₃₃*) → *S₂* (*port₂*) → *host₃*” can be identified on blockchain, which is generated by the detection algorithm in Algorithm 1 based on the corresponding blacklists. We also inspect the flow table entries on switches *S₁* and *S₂*, and the result shows that only *S₁* in source domain installs defense flow table entry “*flow_id* = 65542, *priority* = 200, *in_port* = 9, *dstmac* = 00: 0c: 29: cf: 76: ca, *dstip* = 192.168.188.123, *actions* = drop” (the *dstmac* and *dstip* are the addresses of *host₃*). The above results prove that our proposed collaborative blockchain-based defense policy is implemented. The DDoS traffic have been intercepted in the source domain.

We also append a comparison experiment to verify the effectiveness of our proposed method; that is, only the attack source switch continuously updates the defense flow table, and the normal traffics aimed to targeted domain will not be discarded. We partially modify the proposed mechanism called Isolated BSD-Guard, in which the detection module is reserved, but the smart contract-based (SC-based) collaborative defense module is removed. The attack is performed again in the third experiment, in which the forged TCPSYN0011 packets are launched from *host₁* to *host₃* with the same intensity. After launching an attack, the same detection process is ongoing under BSD-Guard and Isolated BSD-Guard. And the rate of *packet_in* decreases after a few seconds. However, in the Isolated BSD-Guard group, both

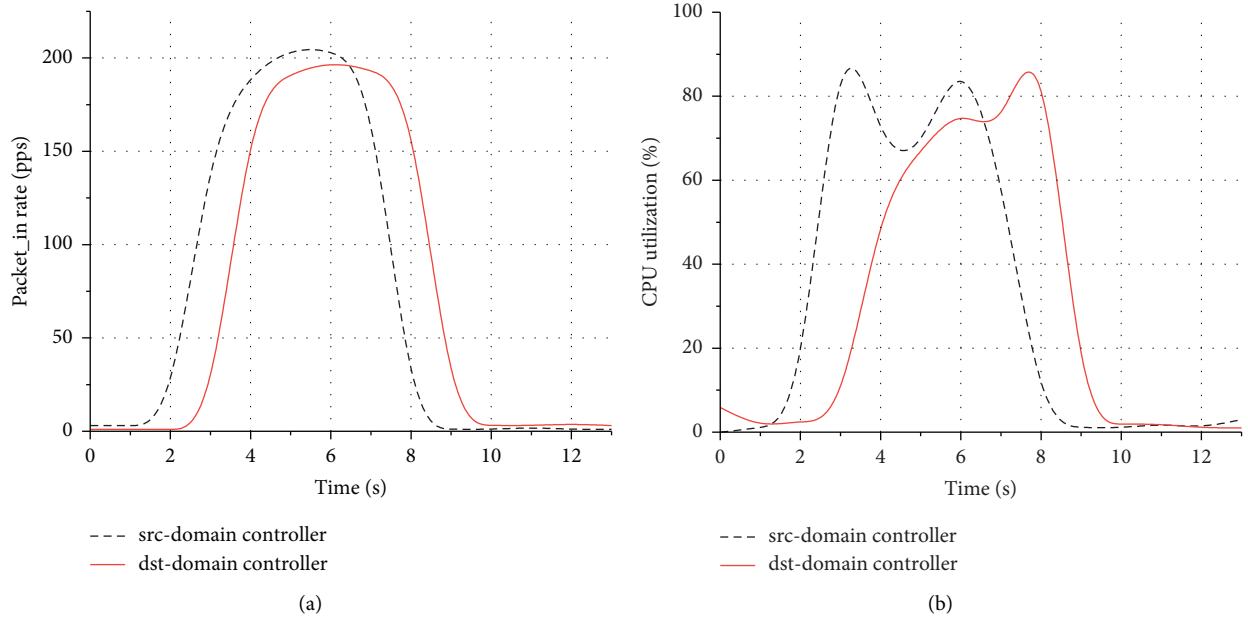


FIGURE 11: The *packet_in* rate and CPU utilization of source and destination domains controller under tcpsyn0011 DDoS attack with BSD-Guard. The change in the attack source domain is ahead of the change in the target domain. (a) *packet_in* rate of source and destination domains controller; (b) CPU utilization of source and destination domains controller.

the switches S_1 and S_2 issue and update defense flow table entries to discard the forged packets targeted to $host_3$. At the same time, we launch the normal TCP SYN request from $host_2$ to $host_3$. The result shows that the request packets cannot reach $host_3$ because of the defense flow table entry on switch S_2 . In contrast, BSD-Guard did not discard request packets from normal user $host_2$ while defending against DoS attacks from $host_1$. The compared results in Table 4 prove that the BSD-Guard can collaborate the suspect traffic information of multidomain SDN, identify the attack path with global view, and mitigate controller targeted DDoS from the source of attack.

We also count the time overhead of our detection and defense process. The detection module consists of state collection and detection algorithm. The collected information is stored to ES database in real time. We divided the total time overhead into four stages, including T1 (searching ES database and computing), T2 (forming black/graylists and uploading to blockchain), T3 (cooperative detecting by smart contract), and T4 (issuing defense strategies). Table 5 demonstrates the time overhead of five groups of experiments, and the average of total time is 675.02 ms, which is mainly occupied by T1 and T2. The millisecond-level block generation speed can meet the requirement of defense and is much faster than 14s in Ethereum [10]. Since what is stored on the blockchain is not the original data of *sFlow* and *packet_in*, but the blacklist and graylist formed by statistical analyzing, the amount of data uploaded on the blockchain is not very large. After the successful defense, the smart contract will also delete the expired data to save space on the blockchain.

TABLE 4: Comparison of BSD-Guard and Isolated BSD-Guard under attack.

Indicators	BSD-Guard	Isolated BSD-Guard
Detect DDoS	Yes	Yes
SC-based defense	Yes	No
Identify attack path	Yes	No
Permit normal flow	Yes	No
Flow table space	Saved	Wasted

TABLE 5: The time overhead during detection and mitigation stages (ms).

Group	1	2	3	4	5
T1	483.37	444.21	307.73	331.55	362.81
T2	276.24	258.21	311.02	206.94	262.71
T3	25.21	23.66	21.55	25.93	27.13
T4	1.50	1.21	0.97	1.63	1.54
Total	786.32	727.29	641.27	566.05	654.19

5.4. Characteristics Analysis. The main objective of BSD-Guard is to provide a collaborative, elastic, lightweight, easy-to-deploy controller targeted DDoS attacks detection and mitigation scheme based on blockchain and smart contract. In this section, we will discuss how our proposed BSD-Guard achieves these characteristics.

- (1) *Easy to Deploy.* The implemented functional modules in the BSD-Guard system are deployed in Docker containers, allowing for rapid deployment and cluster scaling. The FISCO platform [34] is employed to provide blockchain, and the official

WeBASE platform provides convenient nodes and contracts management function.

- (2) *Collaborative Detecting and Defending*. Multiple controllers can share topology and threat information on the trusted blockchain, on which information can not be tampered by malicious attackers. Collaborative defense makes forged attack traffic discarded at the source switch, which reduces the defense overhead of subsequent switches. And the formation of attack path helps the adoption of more precise defense strategies.
- (3) *Precise and Elastic Defending*. The defense policies are established based on the blacklists and graylists stored on blockchain. The generation of defense flow table entries is determined by the characteristics of detected attack traffic, which can avoid dropping of normal traffic. Meanwhile, the duration of defense flow table depends on the real-time load of controller, which makes defense more elastic.
- (4) *Lightweight*. The system employs a private blockchain that does not consume additional gas and does not affect the performance of ONOS controller. The collaborative defense policies save flow table space on hardware switches. Compared with machine learning based detection algorithms, in which the features extracting and data training increase the complexity, our proposed BSD-Guard system is more lightweight.

6. Conclusion

In this paper, we proposed BSD-Guard, a collaborative and elastic blockchain-based detection and mitigation framework to protect SDN against controller targeted DDoS attack. BSD-Guard consists of the secure middle plane and blockchain. The secure middle plane can collect traffic information from data planes, including *sFlow* and *OpenFlow*. The blockchain stores and shares the blacklists and graylists via smart contracts and makes global defense strategies. We design two types of detection and mitigation mechanisms under the intradomain and cross-domain scenarios. We deploy BSM-Guard on the physical environment to verify the effectiveness of our proposed framework. Three groups of experiments have been conducted to verify the system's defense abilities against various types of DDoS attacks. The experimental results indicate that BSD-Guard can detect DDoS attacks with global view, identify the attack path, and install precise defending flow table entries on the near-attack switches. The SDN controller can be well protected and normal service traffic will not be affected by defense policy. Compared with controller clusters, the introduction of blockchain solves the problem of threat sharing among multiple controllers and achieves rapid response and mitigation of DDoS attacks against controllers within an acceptable time and space range.

Data Availability

The experiment data of BSD-Guard are uploaded at <https://github.com/SeuSQ/BSD-Guard/issues/1#issue-1092568880>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1804604 and in part by the General Program of the National Natural Science Foundation of China under Grant 62172093.

References

- [1] A. Abdou, P. C. Van Oorschot, and T. Wan, "Comparative analysis of control plane security of sdn and conventional networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3542–3559, 2018.
- [2] L. F. Eliyan and R. Di Pietro, "Dos and ddos attacks in software defined networks: a survey of existing solutions and research challenges," *Future Generation Computer Systems*, vol. 122, pp. 149–171, 2021.
- [3] M. Imran, M. H. Durad, F. A. Khan, and A. Derhab, "Toward an optimal solution against denial of service attacks in software defined networks," *Future Generation Computer Systems*, vol. 92, pp. 444–453, 2019.
- [4] M. Essaid, D. Kim, S. H. Maeng, S. Park, and H. T. Ju, "A collaborative ddos mitigation solution based on ethereum smart contract and rnn-lstm," in *Proceedings of the 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–6, IEEE, September 2019.
- [5] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, "Insdn: a novel sdn intrusion dataset," *IEEE Access*, vol. 8, Article ID 165263, 2020.
- [6] O. E. Tayfour and M. N. Marsono, "Collaborative detection and mitigation of ddos in software-defined networks," *The Journal of Supercomputing*, vol. 77, no. 11, Article ID 13166, 2021.
- [7] Z. Abou El Houda, A. S. Hafid, and L. Khoukhi, "Cochain-SC: an intra- and inter-domain ddos mitigation scheme based on blockchain using SDN and smart contract," *IEEE Access*, vol. 7, Article ID 98893, 2019.
- [8] Z. Abou El Houda, A. Hafid, and L. Khoukhi, "Co-iot: a collaborative ddos mitigation scheme in iot environment based on blockchain using sdn," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, Waikoloa, HI, USA, December 2019.
- [9] B. Rodrigues, T. Bocek, A. Lareida, D. Hausheer, S. Rafati, and B. Stiller, "A blockchain-based architecture for collaborative ddos mitigation with smart contracts," in *Proceedings of the IFIP International Conference on Autonomous Infrastructure, Management and Security*, pp. 16–29, Springer, Zurich, Switzerland, July 2017.
- [10] J. Dheeraj and S. Gurubharan, "Ddos mitigation using blockchain," *International Journal of Research in Engineering, Science and Management*, vol. 1, no. 10, pp. 622–626, 2018.

- [11] X. You, Y. Feng, and K. Sakurai, "Packet in message based ddos attack detection in sdn network using openflow," in *Proceedings of the 15th International Symposium on Computing and Networking (CANDAR)*, pp. 522–528, IEEE, Aomori, Japan, November 2017.
- [12] X. Huang, X. Du, and B. Song, "An effective ddos defense scheme for sdn," in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Paris, France, May 2017.
- [13] R. F. Fouladi, O. Ermiş, and E. Anarim, "A ddos attack detection and defense scheme using time-series analysis for sdn," *Journal of Information Security and Applications*, vol. 54, Article ID 102587, 2020.
- [14] W. Chen, S. Xiao, L. Liu, X. Jiang, and Z. Tang, "A ddos attacks traceback scheme for sdn-based smart city," *Computers & Electrical Engineering*, vol. 81, Article ID 106503, 2020.
- [15] K. Bhushan and B. B. Gupta, "Distributed denial of service (ddos) attack mitigation in software defined network (sdn)-based cloud computing environment," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 5, pp. 1985–1997, 2019.
- [16] M. Hassan, D. Mahmood, Q. Shaheen, R. Akhtar, W. Changda, and S-dps, "An sdn-based ddos protection system for smart grids," *Security and Communication Networks*, vol. 2021, Article ID 6629098, 19 pages, 2021.
- [17] B. H. Lawal and A. Nuray, "Real-time detection and mitigation of distributed denial of service (ddos) attacks in software defined networking (sdn)," in *Proceedings of the 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, Izmir, Turkey, May 2018.
- [18] C. Kumar, B. P. Kumar, A. Chaudhary et al., "Intelligent ddos detection system in software-defined networking (sdn)," in *Proceedings of the IEEE International Conference on Electronics, Computing and Communication Technologies (CONECT)*, pp. 1–6, IEEE, Bangalore, India, July 2020.
- [19] Y. Lu and M. Wang, "An easy defense mechanism against botnet-based ddos flooding attack originated in sdn environment using sflow," in *Proceedings of the 11th International Conference on Future Internet Technologies*, pp. 14–20, ACM, Nanjing, China, June 2016.
- [20] K.-Y. Chen, S. Liu, Y. Xu et al., "Sdnshield: nfv-based defense framework against ddos attacks on sdn control plane," *IEEE/ACM Transactions on Networking*, vol. 30, 2021.
- [21] S. Y. Mehr and B. Ramamurthy, "An svm based ddos attack detection method for ryu sdn controller," in *Proceedings of the 15th international conference on emerging networking experiments and technologies*, pp. 72–73, ACM, Orlando, FL, USA, December 2019.
- [22] J. Cui, J. Zhang, J. He, H. Zhong, and Y. Lu, "Ddos detection and defense mechanism for sdn controllers with k-means," in *Proceedings of the IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, pp. 394–401, IEEE, Leicester, UK, December 2020.
- [23] V. Deepa, K. M. Sudar, and P. Deepalakshmi, "Detection of ddos attack on sdn control plane using hybrid machine learning techniques," in *Proceedings of the International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 299–303, IEEE, Tirunelveli, India, December 2018.
- [24] B. Nugraha and R. N. Murthy, "Deep learning-based slow ddos attack detection in sdn-based networks," in *Proceedings of the IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 51–56, IEEE, Leganes, Spain, November 2020.
- [25] Y. Xu, Y. Yu, H. Hong, and Z. Sun, "Ddos detection using a cloud-edge collaboration method based on entropy-measuring som and kd-tree in sdn," *Security and Communication Networks*, vol. 2021, Article ID 5594468, 16 pages, 2021.
- [26] R. M. A. Ujjan, Z. Pervez, K. Dahal, A. K. Bashir, R. Mumtaz, and J. González, "Towards sflow and adaptive polling sampling for deep learning based ddos detection in sdn," *Future Generation Computer Systems*, vol. 111, pp. 763–779, 2020.
- [27] T.-K. Luong, T.-D. Tran, and G.-T. Le, "Ddos attack detection and defense in sdn based on machine learning," in *Proceedings of the 7th NAFOSTED Conference on Information and Computer Science (NICS)*, pp. 31–35, IEEE, Ho Chi Minh City, Vietnam, November 2020.
- [28] K. Nishizuka, L. Xia, J. Xia, D. Zhang, L. Fang, and C. Gray, *Interorganization Cooperative Ddos protection Mechanism*, Internet-Draft, 2016.
- [29] J. Steinberger, B. Kuhnert, A. Sperotto, H. Baier, and A. Pras, "Collaborative ddos defense using flow-based security event information," in *Proceedings of the NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 516–522, IEEE, Istanbul, Turkey, April 2016.
- [30] U. Javaid, A. K. Siang, M. N. Aman, and B. Sikdar, "Mitigating lot device based ddos attacks using blockchain," in *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, pp. 71–76, New York, NY, USA, 2018.
- [31] Z. Shao, X. Zhu, A. M. M. Chikuvanyanga, and H. Zhu, "Blockchain-based sdn security guaranteeing algorithm and analysis model," in *Proceedings of the International Conference on Wireless and Satellite Systems*, pp. 348–362, Springer, Harbin, China, January 2019.
- [32] S. Gao, Z. Peng, B. Xiao, A. Hu, Y. Song, and K. Ren, "Detection and mitigation of dos attacks in software defined networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1419–1433, 2020.
- [33] K. Giotis, M. Apostolaki, and V. Maglaris, "A reputation-based collaborative schema for the mitigation of distributed attacks in sdn domains," in *Proceedings of the NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 495–501, IEEE, Istanbul, Turkey, April 2016.
- [34] Fisco-bcos, "Fisco-bcos-documentation," 2021, https://fisco-bcos-documentation.readthedocs.io/zh_CN/latest/.

Research Article

A Noninteractive Multireplica Provable Data Possession Scheme Based on Smart Contract

Zhengwen Li ¹, Yang Xin,¹ De Zhao ², and Yixian Yang¹

¹School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China

²School of Information Engineering, Beijing Institute of Graphic Communication, Beijing 102600, China

Correspondence should be addressed to Zhengwen Li; lizhengwen@bupt.edu.cn

Received 19 November 2021; Revised 25 January 2022; Accepted 8 March 2022; Published 6 April 2022

Academic Editor: Xin-Yi Huang

Copyright © 2022 Zhengwen Li 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 explosive growth of data, cloud storage has become a widely used storage method. To protect the integrity and availability of data in cloud storage systems, multireplica provable data possession has gradually become a research hotspot. This paper uses smart contracts to replace traditional third-party auditor (TPA) and proposes a noninteractive multireplica provable data possession scheme based on smart contracts, making the verification process public, immutable, traceable, and able to be carried out periodically and automatically. This paper introduces the concept of noninteractivity to reduce the transaction fees caused by the frequent operation of blockchain in the verification process. By stipulating payment rules in the smart contract, we can ensure the fairness of all parties. Finally, we give the correctness proof of the scheme and the security proof in the random oracle model, comparing it with other schemes and verifying the practicability of our scheme through experiments.

1. Introduction

In recent years, with the continuous development of the Internet of Things, big data, artificial intelligence, mobile Internet, and other fields, the amount of data generated by people has increased explosively. According to IDC [1], the total amount of data in the world will increase from 33 ZB in 2018 to 175 ZB in 2025, and the data will become a precious strategic resource. Cloud storage has gradually become the data storage trend due to its low cost, flexible scalability, and anytime and anywhere access. The most popular products are Amazon S3, Google Drive, Microsoft Azure, Dropbox, Alibaba Cloud, etc.

Generally, after uploading local data to the cloud server using cloud storage services, users will delete the original data to save local storage resources. Due to the separation of cloud storage data ownership and physical control, users cannot timely understand the actual storage status of data, which makes the availability and integrity of data one of the most concerning issues of cloud storage security for users.

Data availability means that users can get data in time when they need it and recover the original data when there is

a certain degree of error in the data. To ensure data availability, multireplica and erasure code are two widely used technologies. Multireplica technology usually stores multiple replicas on multiple servers. If a replica of data is damaged, it can be recovered using replicas of other data centers. Erasure code is a coding technology, which uses redundant blocks to provide fault tolerance. When part of the data is damaged, it can be reconstructed by coding. Compared with erasure code, multireplica technology uses more storage space, but its implementation is more straightforward and consumes less computing resources, so it is more widely used.

Data integrity means that specific data remain completely unchanged during storage or transmission. To ensure data integrity, Provable Data Possession (PDP) and Proofs of Retrievability (POR) are two widely used methods. PDP is used mainly to complete data integrity verification quickly, and POR is used to ensure data integrity due to its ability to recover data. It consumes additional computing resources and storage space.

Cloud storage service providers (CSP) are not entirely credible. User data may be damaged and unavailable due to

power interruption, hacker attacks, and software and hardware failures, and even some CSPs deliberately tamper, destroy, and delete user data for some purpose. To avoid downloading data before finding that the data are unavailable, users should periodically check the data integrity in the CSP. Combined with the current situation that CSP has widely adopted multireplica technology, doing multi-replica provable data possession safely and efficiently has become a research hotspot in recent years.

1.1. Related Work. Ateniese et al. [2] first proposed the PDP scheme, which obtains the probability of data integrally possessed by the server through random sampling of data blocks, allowing users to check whether the server has stored the entire data without downloading all data. It uses the homomorphic verification tag based on RSA to reduce the computational overhead and improve the efficiency of integrity verification. This scheme is also the first to support public verification, which can meet the needs of third-party verification. Around the same time, Juels and Kaliski [3] proposed the POR scheme, which is based on the sentinel mechanism and can restore damaged data while providing integrity verification. Since the number of sentinels is fixed and the verification consumes several sentinels each time, the scheme has finite verification times. In addition, this paper presents a formal security definition of integrity verification for the first time, which is instructive for follow-up research. Shacham and Waters [4] proposed two POR schemes based on BLS signature and pseudorandom functions. They presented complete proofs of security of the two schemes under the random oracle model and the standard model through the interactive analysis of a series of games. Wang et al. [5] proposed a PDP scheme based on the BLS signature, which supports public verification and dynamic data update by constructing Merkle hash trees of the data block tag authenticator. To fully ensure data security and conserve the computational resources of users, Wang [6] introduced the TPA to complete the verification work and propose a public verification scheme supporting privacy preserving by combining homomorphic linear authenticator and random masking technique, which can batch process multiple verification tasks.

To satisfy users' demands for data availability, CSP duplicates the data into corresponding replicas and stores multiple replicas on multiple servers. For multireplica provable data possession, Curtmola et al. [7] proposed an MR-PDP scheme to reduce the overhead of integrity verification of all copies to roughly the same as a single copy. Unfortunately, this scheme only supports private verification. Hao and Yu [8] proposed a multireplica remote data possession check protocol with public verifiability by combining a homomorphic verification tag and BLS signatures. Wei [9] proposed an efficient dynamic replicated data possession verification scheme, which uses the fully homomorphic encryption (FHE) algorithm to generate multiple replicas and resist forgery, replacement, and replay attacks. Ya-Xing [10] proposed a new multiuser and multiple-replica provable data possession scheme. The scheme

adopts random mask technology to process ciphertext to ensure data privacy. It adopts a multibranch authentication tree to improve the efficiency of data block signature, which can support dynamic data update operation and batch audit. Peng et al. [11] proposed an identity-based multiple-replica data integrity checking scheme (EDID-MRPDP), introducing a new Homomorphic Verifiable Tag (HVT) structure and a new Compressed Authentication Array (CAA) data structure, which can simultaneously and efficiently conduct batch authentication for multiple owners and cloud servers. Yu et al. [12] proposed a dynamic multiple-replica auditing scheme, which can simultaneously verify the integrity and geographic location of the replica data of cloud users by introducing an Indexed Merkle Hash Tree (IMHT), and the problem of the excessive overhead of the existing Merkle hash tree can be reduced.

Most of the above integrity verification schemes assume that TPA is credible, which is bold and dangerous. If the auditor colluded with the CSP or the attacker, the provable data integrity provided by the auditor would become unreliable [13–17]. Meanwhile, TPA is also faced with a single point of failure and performance limitations. Fortunately, the emergence of blockchain technology provides a new way to solve these problems because the essence of blockchain technology is a mutual trust mechanism based on mathematical algorithms. In addition, blockchain has the characteristics of decentralization, openness, transparency, tamperproof, and traceability, which coincide with the requirements of data integrity audit. Nowadays, more and more scholars have begun to combine blockchain technology to research provable data integrity.

Some schemes [18–20] only take advantage of the openness, transparency, and tamper-proof characteristics of the blockchain to store verification logs on the blockchain but do not eliminate the threat of malicious TPA. Huang et al. [21] proposed a collaborative auditing blockchain framework for cloud data storage by using all consensus nodes substituting the single third-party auditor to execute auditing delegations and record them permanently, but not for multiple replicas. Xu [22] proposed a decentralized and arbitrable data auditing scheme based on blockchain. It mainly uses the communicative hash technique to randomly verify the integrity of a group of data blocks to probabilistically verify the integrity of all data, and it completes the information interaction in the verification process through blockchain transactions. It uses smart contracts to realize the adjudication mechanism without TPA. However, the scheme is too idealized, almost every data block needs to participate in the verification, and the interaction process will produce a large number of blockchain transaction costs, which makes the scheme very impractical. Chen et al. [23] proposed the first decentralized system BOSSA for proofs of data retrievability and replication. Since the blockchain cannot actively issue challenges and reacts based on received transactions, this paper proposes a time-restricted proof forcing the cloud to prove data availability. In addition, the scheme is aimed at the decentralized storage network, where other nodes store replicas, so the replicas must be encoded and encrypted to ensure privacy and reliability. Fan et al.

[24] used a smart contract to replace TPA and proposed a decentralized audit scheme on Ethereum called Dredas; anyone can obtain audit results from Ethereum without worrying about semihonest TPA. This solution uses a smart contract and ether to propose a deposit mechanism to pay audit fees and punish malicious behavior. At the same time, the solution also supports batch audit and dynamic data audit. However, the scheme does not consider the case of multiple replicas, and a large amount of information needs to be stored in the contract in the audit process, which has the problem of high interaction cost. Wang et al. [25] used blockchain to replace TPA and designed a blockchain-based fair payment smart contract for a public audit of cloud storage. This contract ensures that CSP needs to submit provable data possession termly. To reduce the number of times of interactions during the execution of the contract, the concept of noninteractive provable data possession was first proposed. Unfortunately, multireplica is not considered. Li et al. [26] proposed a decentralized storage framework supporting provable data possession based on blockchain—IntegrityChain, which can simultaneously protect data confidentiality, integrity, and availability by using pseudorandom function and multireplica technology. However, all interactions in this scheme are completed through transactions of blockchain, and gas is consumed in each step, so the transaction cost is significantly increased. Chen et al. [27] proposed a decentralized outsourcing storage system that supports dynamic provable data possession based on the blockchain. The applicable scenario is P2P storage network. All storage and audit behaviors will generate transactions, and then blockchain is used to record all transactions. The scheme utilizes smart contract to support public verification, ensures fairness of all parties by deposit mechanism, and takes advantage of an authenticated data structure (ADS) called rank-based Merkle hash tree to support updating operations. However, the scheme is not lightweight enough, and additional Merkle tree structure and auxiliary verification information need to be stored in the transaction, which has a great burden on the operation of blockchain.

Existing schemes do not fully account for the transaction fees on the blockchain, which would be higher if the data were to be manipulated in a complex manner. In addition, the storage capacity of each block is so small that it is impossible to store large amounts of data or complex data structures in practice. Therefore, considering the limited storage capacity of blocks and the high transaction fees caused by frequent interactions, we improved the multireplica provable data possession protocol based on the BLS signature and homomorphic authentication tag and proposed a noninteractive lightweight scheme combined with a smart contract.

1.2. Our Contribution. Our contributions are summarized as follows.

- (1) A noninteractive multireplica provable data possession protocol NI-MR-PDP is designed to support public verification, batch processing, and privacy

preserving; all parties in the system do not need to carry out challenge-response interaction. A series of games are constructed for interactive analysis to prove that the protocol is safe in the random oracle model.

- (2) A noninteractive multireplica provable data possession scheme based on smart contracts is proposed. Deploying smart contracts on blockchain to eliminate the dependence on untrusted TPA can automatically verify data integrity openly, transparently, and periodically. According to the content of the smart contract, the rights and obligations of the participating parties are stipulated. Once the conditional contract is triggered, automatic execution of the contract can protect the legitimate rights of all parties and reduce the settlement cost of disputes. After deploying the contract, the parties in the system do not need to interact, which helps the consensus nodes in the blockchain to efficiently implement the smart contract.
- (3) A series of games are constructed for interactive analysis to prove that the scheme is safe under the random oracle model. Experiments show that the scheme is practical and has good efficiency.

1.3. Organization. The rest of the paper is organized as follows. Section 2 recalls some preliminaries used in our scheme. Section 3 defines the model of our scheme, gives the formal definition, and presents the concrete construction. Section 4 provides the security proof of the protocol. Section 5 evaluates the performance of our scheme. Finally, we give a conclusion in Section 6.

2. Preliminaries

2.1. Bilinear Map. Let G_1 , G_2 , and G_T be multiplicative cyclic groups of prime order p , g_1 a generator of G_1 , and g_2 a generator of G_2 . A bilinear map $e: G_1 \times G_2 \rightarrow G_T$ has the following properties.

- (1) Bilinear: $\forall u \in G_1, v \in G_2$ and $\forall a, b \in \mathbb{Z}_p$, there is $e(u^a, v^b) = e(u, v)^{ab}$
- (2) Nondegenerate: $e(g_1, g_2) \neq 1$
- (3) Computable: $\forall u \in G_1, v \in G_2$, and there is an efficient algorithm to calculate $e(u, v)$

2.2. BLS Signature. Dan Boneh [28] proposed the BLS signature scheme, which uses bilinear mapping to verify the elements in the elliptic curve group. The core idea is to verify the correctness of the digital signature while protecting the user's private key from being leaked. The signature length of BLS is shortened to 160 bits, which is shorter than a typical signature at the same security level.

Let the signature algorithm be based on bilinear mapping $e: G_1 \times G_2 \rightarrow G_T$, where G_1 , G_2 , and G_T are multiplicative cyclic groups of prime order p , g_1 is a generator of G_1 , and g_2 is a generator of G_2 .

BLS signature algorithm includes three algorithms: key generation algorithm, signature algorithm, and verification algorithm. The specific description is as follows.

- (1) SKg: it is used to generate a pair of the public and secret keys of the signature scheme. The user randomly selects a value $x \in Z_p$ as the secret key, and the corresponding public key is $g_2^x \in G_2$.
- (2) SSing: it is used to complete the signature of the message. Given a secret key x and message $m \in \{0, 1\}^*$, compute the hash of the message $h = H(m)$, $h \in G_1$ and output the signature $\sigma = h^x$, $\sigma \in G_1$.
- (3) SVerify: it is used to verify the validity of the signature. Given a message m , signature σ , and public key g_2^x , check whether $e(\sigma, g) = e(h, g_2^x)$ holds. If it holds, the signature is valid. Otherwise, it is invalid.

2.3. Blockchain and Smart Contract. In 2008, Satoshi Nakamoto [29] proposed the concept of bitcoin, and blockchain as the core technology of bitcoin was proposed for the first time. Blockchain is essentially a chained data structure that combines data blocks in chronological order and is a tamper-proof and unforgeable distributed ledger guaranteed by cryptography.

In the blockchain, data are permanently stored in blocks, and blocks are generated one by one in chronological order and connected into a chain. As shown in Figure 1, each block contains a block header and a block body. The block header contains the previous block's hash value, version number, random value, timestamp, Merkle root hash, and difficulty value. The block body contains all transaction information generated during the block creation process. Each block in the blockchain is identified by a hash value obtained by the secondary SHA256 hash calculation of the block header. Each block can find its previous block by the previous block hash value contained in its block header. Any change to a block on the blockchain will lead to a series of changes in subsequent blocks. Distributed nodes synchronously update the hash chain by running a consensus protocol. Therefore, blockchain has the characteristics of decentralization, transparency, openness, tamper-resistant, and traceability.

In 1997, the smart contract was formally proposed by Nick Szabo [30], which is an electronic quantitative trading protocol for contract terms in reality. The essence of a smart contract is a piece of code running on the blockchain. The logic of the code defines the content of the smart contract. After the smart contract is successfully deployed, once the agreed rules are met, the contract content can be automatically executed without the participation of intermediaries, and no one can prevent it from running. We can say that blockchain provides a trusted execution environment for smart contracts, and smart contracts extend blockchain's application. The smart contract has been applied in many fields, such as electronic voting [31] and insurance [32], and has broad prospects.

3. Our Scheme

3.1. System Model. The system model of a noninteractive multireplica provable data possession scheme includes three entities: data owner, cloud storage service provider, and verifier (see Figure 2).

- (1) Data Owner (DO): cloud storage service users choose to pay a certain fee to store their data in remote servers of the cloud storage service provider to save local storage costs and use data flexibly and conveniently.
- (2) Cloud Storage Service Provider (CSP): it is composed of multiple replica servers, which adopt multireplica technology to improve data availability and provide computing resources, storage resources, and network bandwidth resources for DO. CSP needs to verify the data integrity of multiple replicas periodically. If the verification fails, it will compensate for a certain fee to DO.
- (3) Verifier: it is the proof verification algorithm executor; because of the public verification of the scheme, theoretically all members of the blockchain can act as verifiers, usually by third-party miners. The verifier obtains a certain reward by executing the smart contract deployed on the blockchain.

3.2. Formal Definition. The scheme is divided into two phases: the setup and audit phases. It consists of five algorithms: KeyGen, ReplicaGen, TagGen, ProofGen, and ProofVerify. Each algorithm is formally defined as follows.

- (1) Key Gen(1^λ) \longrightarrow (pk, sk): key generation algorithm is run by DO. The algorithm's input is a security parameter λ , and the output is public key pk and secret key sk .
- (2) Replica Gen(F) \longrightarrow $\{F_d\}$: replica generation algorithm is run by DO. The algorithm's input is ciphertext F , and the output is t different replicas $\{F_d\}, 1 \leq d \leq t$.
- (3) Tag Gen(sk, F_d) \longrightarrow (τ_d, ψ): tag generation algorithm is run by DO. The algorithm's input is the secret key sk and replica F_d , and the output is the tag of replica τ_d and the tag set of blocks ψ .
- (4) Proof Gen($\theta, F_d, \tau_d, \psi$) \longrightarrow P_d : proof generation algorithm is run by CSP. The algorithm's input is public state information θ , replica F_d , the tag of replica τ_d , and the tag set of blocks ψ , and the output is proof $P_d, 1 \leq d \leq t$.
- (5) Proof Verify(pk, θ, P) \longrightarrow (SUCCESS, FALSE): proof verification algorithm is run by the verifier. The algorithm's input is public key pk , public state information θ , and proof $P = \{P_d\}$. If the verification succeeds, the output is SUCCESS, and if the verification fails, the result is FALSE.

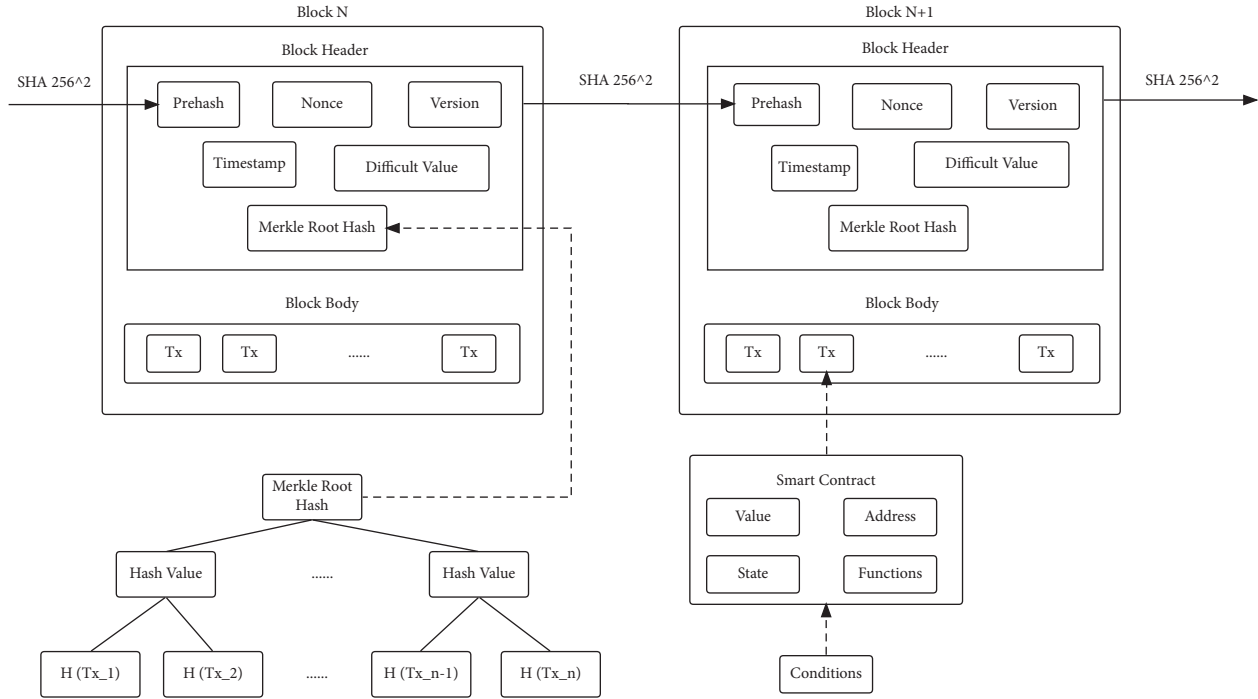


FIGURE 1: Blockchain structure.

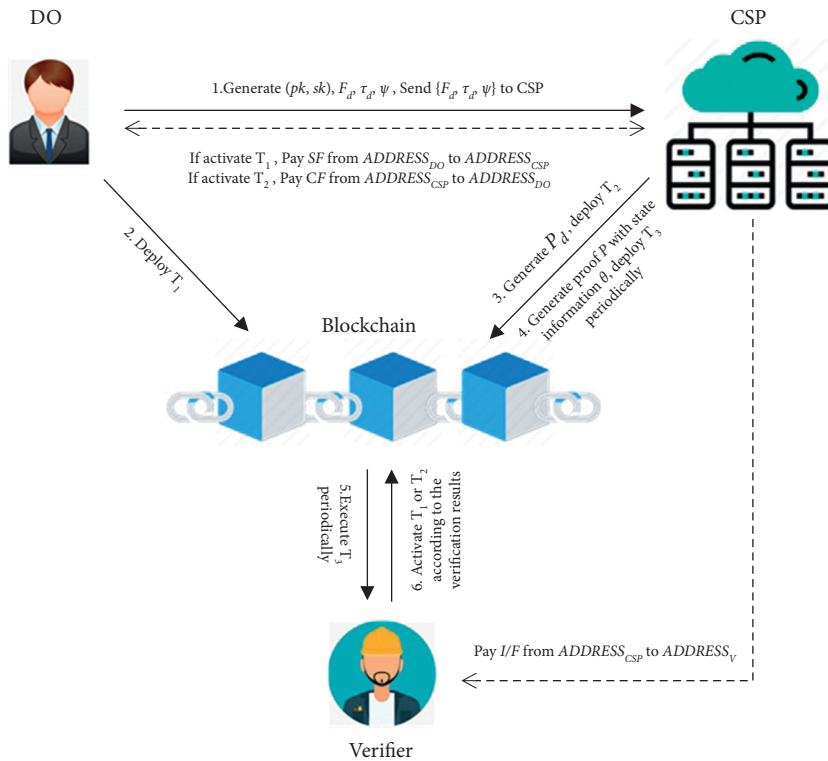


FIGURE 2: System model.

3.3. *Scheme Implementation.* In this part, we first introduce the noninteractive multireplica provable data possession (NI-MR-PDP) protocol. Then we propose our noninteractive multireplica data possession scheme combined with smart contract technology.

3.3.1. *NI-MR-PDP.*

(1). *Setup Phase.* Let G and G_T be multiplicative cyclic groups of prime order p and g a generator of G , and there is a bilinear map $e: G \times G \rightarrow G_T$. Two hash functions are $H(\cdot): \{0, 1\}^* \rightarrow G$ and $h(\cdot): \{0, 1\}^* \rightarrow Z_p$. Two

pseudorandom functions $\text{are}\alpha(\cdot): \{0, 1\}^* \rightarrow \{0, 1\}^*$ and $\beta(\cdot): \{0, 1\}^* \rightarrow [1, n]$. Let the number of challenge blocks be an integer $c, 1 \leq c \leq n$.

KeyGen: it selects λ as the security parameter, randomly generates a pair of public and secret keys $(spk, ssk) \leftarrow SKg$ for signature, and computes $v \leftarrow g^{ssk}$. The public key is $pk = (spk, v)$ and the secret key is $sk = ssk$.

ReplicaGen: DO encrypts the file to get the ciphertext F , divided into n blocks with the same size and expressed as $F = \{f_i\}, 1 \leq i \leq n$. To prevent the adversary from using files of different replica servers to restore the complete F , DO needs to generate a unique and distinguishable replica file. By adding random values, t various replicas $F_d = \{m_{d,i}\}$ are generated, where $m_{d,i} = f_i + r_{d,i}, r_{d,i} = \alpha(d \| i), 1 \leq d \leq t, 1 \leq i \leq n$. The more replicas, the higher reliability of the data, and the corresponding fee charged by CSP will increase.

TagGen: DO randomly selects $\text{name} \leftarrow Z_p$ and t values $u_1, u_2, \dots, u_t \leftarrow G$ and calculates the tag $\sigma_{d,i} = (H(\text{name} \| i) \cdot u_d^{m_{d,i}})^{sk}$ of each block $m_{d,i}$ in the replica. Due to the aggregation of the BLS signature, the tags of the same subscript blocks of different replicas F_d can be aggregated into $\sigma_i = \prod_{d=1}^t \sigma_{d,i}$, and the tag set of blocks is $\psi = \{\sigma_i\}$. Let $\tau_{o,d} = \text{name} \| n \| d \| u_1 \| \dots \| u_t$, and the tag of each replica is $\tau_d = \tau_{o,d} \| S\text{Sign}_{sk}(\tau_{o,d})$.

(2). **Audit Phase.** At this phase, the Verifier does not need to randomly select the challenge set to challenge CSP, like the traditional PDP scheme. Instead, the CSP uses the current public state information θ as the input of the pseudorandom function to simulate the challenge set generation process. In this way, the Verifier can generate a challenge set by itself to meet the requirement of no interaction.

In our scheme, θ should be publicly available and not controlled by the CSP while changing over time. Considering the blockchain structure, the timestamp or hash value of the previous block in the block header can meet the above requirements and be used as θ .

ProofGen: CSP selects an appropriate integer $c, 1 \leq c \leq n$. For $\forall j \in [1, c]$ it computes $s_j \leftarrow \beta(\theta \| j)$; we can get $I = \{s_1, s_2, \dots, s_c\}$. For $\forall i \in I$, it calculates $V_i \leftarrow h(\theta \| j)$, so the challenge set is $Q = \{(i, V_i), i \in I$. Each replica server of CSP generates the corresponding proof for its stored replica and computes $\mu_d = \sum_{(i, V_i) \in Q} V_i \cdot m_{d,i}$ and $\sigma = \prod_{(i, V_i) \in Q} \sigma_i^{V_i}$; the proof of replica F_d is $P_d = \{\theta, \tau_d, \mu_d, \sigma\}, 1 \leq d \leq t$.

ProofVerify: Verifier first compares the state information to verify the correctness of the public state information θ , and if it fails, it returns FALSE, and if it succeeds, it computes $I = \{\beta(\theta \| 1), \beta(\theta \| 2), \dots, \beta(\theta \| c)\}$ and $V_i \leftarrow h(\theta \| i)$ and then gets the challenge set $Q = \{(i, V_i), i \in I$.

Then spk is used to verify the tag of the replica τ_d . If it fails, it returns to FALSE. If it succeeds, it returns name, n, d and u_1, u_2, \dots, u_t .

Finally, we check the equation $e(\sigma, g) \stackrel{?}{=} e(\prod_{(i, V_i) \in Q} \prod_{d=1}^t H(\text{name} \| i)^{V_i} \cdot u_d^{\mu_d}, v)$. If the equation holds, output SUCCESS. Otherwise, return FALSE.

It is easy to prove the correctness of the scheme because $v = g^{sk}, \sigma_i = \prod_{d=1}^t \sigma_{d,i}, \sigma_{d,i} = (H(\text{name} \| i) \cdot u_d^{m_{d,i}})^{sk}, \mu_d =$

$\sum_{(i, V_i) \in Q} V_i \cdot m_{d,i}$ and $\sigma = \prod_{(i, V_i) \in Q} \sigma_i^{V_i}$; the equation is as follows:

$$\begin{aligned} e(\sigma, g) &= e\left(\prod_{(i, V_i) \in Q} \sigma_i^{V_i}, g\right) \\ &= e\left(\prod_{(i, V_i) \in Q} \left(\prod_{d=1}^t \sigma_{d,i}\right)^{V_i}, g\right) \\ &= e\left(\prod_{(i, V_i) \in Q} \left(\prod_{d=1}^t (H(\text{name} \| i) \cdot u_d^{m_{d,i}})^{sk}\right)^{V_i}, g\right) \\ &= e\left(\prod_{(i, V_i) \in Q} \prod_{d=1}^t H(\text{name} \| i)^{V_i} \cdot u_d^{\mu_d \cdot V_i}, g^{sk}\right) \\ &= e\left(\prod_{(i, V_i) \in Q} \prod_{d=1}^t H(\text{name} \| i)^{V_i} \cdot u_d^{\mu_d}, v\right). \end{aligned} \quad (1)$$

3.3.2. Smart Contract Scheme. In the traditional cloud storage system, DO needs to pay a certain fee to CSP to purchase storage space. Once the DO data is unavailable or tampered with, it is challenging to obtain economic compensation for data rights. On the one hand, DO will no longer store the original data locally, and the proof process will become arduous. On the other hand, the laws and regulations of various countries on data security are not necessarily complete, especially in the case of transnational disputes. Moreover, legal litigation usually means extra money and time costs.

The emergence of smart contracts brings dawn to solve these problems. Since the smart contract has the characteristics of tamper-resistant and automatic triggering, once the contract content is agreed by all parties, as long as the contract conditions are met, the contract results will be implemented immediately. No one can change the contract content again.

Therefore, we design a noninteractive multireplica provable data possession scheme based on smart contracts in the cloud storage system. By deploying smart contracts on the blockchain, we provide tamper-resistant multireplica integrity verification for all parties and guarantee fair payment through the deposit mechanism. A consensus mechanism needs to be used to fight against dishonest verifiers and ensure the correctness of the verifier's smart contract execution. This article does not discuss that in depth.

The flow logic of the scheme is shown in Figure 3, which is described as follows.

- (1) DO, CSP, and Verifier register on the blockchain to obtain public-secret key pairs and account addresses ADDRESS_{DO} , ADDRESS_{CSP} , and ADDRESS_V

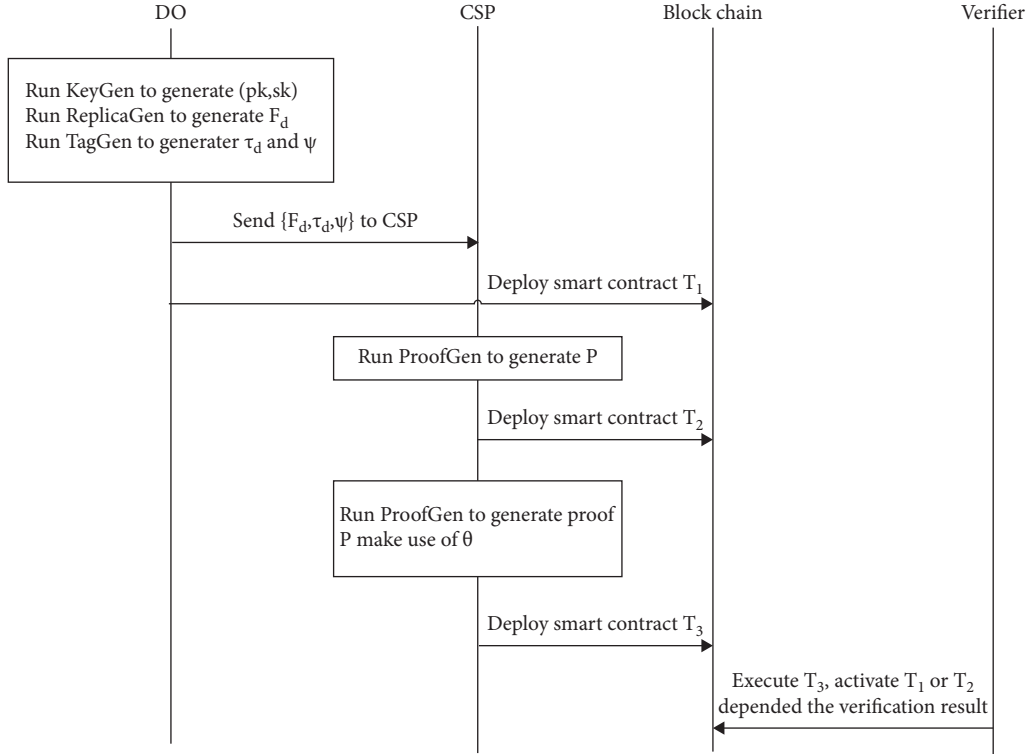


FIGURE 3: The flow logic of the scheme.

respectively. The public-secret key pair is used for signature and verification on the blockchain. The public key usually generates the account address to identify the identity and conduct transactions. DO and CSP need to pay a certain deposit respectively to ensure the smooth completion of subsequent transactions.

- (2) DO runs the algorithm KeyGen to generate public-secret key pair (pk, sk) , runs the algorithm ReplicaGen to generate t different replicas F_d of file F , and runs the algorithm TagGen to generate the tag of replica τ_d and the tag set of blocks ψ .
- (3) DO uploads $\{F_d, \tau_d, \psi\}$ to each replica server of CSP, generates a smart contract T_1 (see Table 1), and deploys it on the blockchain. The smart contract includes the basic information of F (file name, file hash, and upload time), transaction information (storage fee, account address of DO, account address of CSP), and the signature of DO. It can ensure that if the CSP completely stores a replica of the file, it can pass the verification, and DO must pay the cost to the CSP in time.
- (4) After receiving $\{F_d, \tau_d, \psi\}$, each replica server of CSP runs the algorithm ProofGen to generate P_d and then sends it to CSP to obtain the proof set $P = \{P_d\}$. CSP generates a smart contract T_2 (see Table 2) and deploys it on the blockchain. The smart contract

TABLE 1: Storage smart contract T_1 .

Smart contract T_1	
File name	FN
File hash	FH
Upload time	UT
Storage fee	SF
Account addresses of DO	ADDRESS _{DO}
Account addresses of CSP	ADDRESS _{CSP}
Signature of DO	Sign _{DO}
Contract content:	
Promise	
{if Proof Verify $(pk, \theta, P) \rightarrow$ SUCCESS	
Pay SF from ADDRESS _{DO} to ADDRESS _{CSP} }	

TABLE 2: Compensation smart contract T_2 .

Smart contract T_2	
File name	FN
File hash	FH
Receiving time	RT
Compensation fee	CF
Account addresses of DO	ADDRESS _{DO}
Account addresses of CSP	ADDRESS _{CSP}
Signature of DO	Sign _{CSP}
Contract content:	
Promise	
{if Proof Verify $(pk, \theta, P) \rightarrow$ FALSE	
Pay CF from ADDRESS _{CSP} to ADDRESS _{DO} }	

includes the basic information of F (file name, file hash, and receiving time), transaction information (compensation fee, account address of DO, and account address of CSP), and the signature of CSP. It can ensure that if the CSP does not store complete replicas and the integrity verification fails, the CSP must pay a certain fee to compensate the DO in time.

- (5) CSP will periodically generate the corresponding proof combined with the current public state information θ , generate a smart contract T_3 (see Table 3), and deploy it on the blockchain. The smart contract includes the basic information of the file (file name, file hash, generation time of evidence, status information, and evidence information), the contract information to be called, transaction information (verification fee, CSP account address, and verifier account address), and the signature of the CSP. The verifier will execute a smart contract to complete multiple-replica data integrity verification for a reward and then activate T_1 or T_2 based on the verification result.

3.4. Brief Summary. We propose a noninteractive multi-replica provable data possession scheme based on smart contract, which not only meets the basic requirements of correctness and security but also has the following characteristics:

- (1) Public verification: the evidence verification algorithm is public and does not need to use the private key, so any third party can obtain a public conclusion about whether the data have integrity.
- (2) Noninteractive: during the whole verification process, DO and CSP do not need to interact with the third-party verifier, and it DO and CSP do not need to remain online all the time, making the operation of the scheme more flexible.
- (3) Batch verification: batch verification can be carried out simultaneously on all replicas. Only one equation needs to be verified, and then it will tell whether all replicas are stored completely.
- (4) Fair payment: the payments of DO, CSP, and Verifier follow the agreed smart contract, which cannot be tampered with, and cannot be denied by anyone.
- (5) Privacy-preserving: during the whole verification process, the relevant information that the Verifier can access of DO is all encrypted files and cannot obtain any knowledge of DO's original file without knowing the secret key.

4. Security Proof

The security of the scheme is defined by formally describing a security game between challenger C and adversary A :

C generates a public-private key pair (pk, sk) by running KeyGen, sends pk to A , and reminds sk for responding to A 's query.

TABLE 3: Verification smart contract T_3 .

Smart contract T_3	
File name	FN
File hash	FH
Proof generated time	PGT
State information	θ
Proof information	P
Storage smart contract	T_1
Compensation smart contract	T_2
Verification fee	VF
Account addresses of CSP	ADDRESS _{CSP}
Account addresses of verifier	ADDRESS _V
Signature of CSP	Sign _{CSP}
Contract content:	
Promise	
{Execute the ProofVerify algorithm if	
Proof Verify $(pk, \theta, P) \rightarrow$ SUCCESS, activate T_1	
if Proof Verify $(pk, \theta, P) \rightarrow$ FALSE, activate T_2	
Pay VF from ADDRESS _{CSP} to ADDRESS _V }	

- (1) A can query replicas by interacting with C . A randomly selects a file F and sends it to C . C generates t different replicas $F_d (1 \leq d \leq t)$ by running the ReplicaGen algorithm and responds to A .
- (2) A can query tags by interacting with C . A randomly selected replica F_d and sends it to C , and C generates the tag of replica τ_d and the tag set of blocks ψ by running the TagGen algorithm and responds to A .
- (3) A generates proof P' according to responses from multiple queries.

Definition 1. The advantage of adversary A in the game is $A \text{ } dv_A = \Pr[\text{Proof Verify}(pk, \theta, P') = \text{SUCCESS}]$. We say A wins the game if $A \text{ } dv_A$ is nonnegligible.

Definition 2. A noninteractive multi-replica provable data possession scheme is secure. If there is an effective extraction algorithm Extr, for any adversary who wins the security game and the output of the proof of file F is P' , the probability that the Extr can recover the replicas $\{F_d\}$ (i.e., $\text{Extr}(pk, \theta, \tau_d, P') = \{F_d\}$) is nonnegligible.

Theorem 1. If the signature algorithm used to generate file tags is existential unforgeability, the computational Diffie-Hellman problem on bilinear groups and the discrete logarithm problem are difficult; then, in the random oracle model, the probability that an adversary which breaks the security of our scheme, through the verification algorithm using proof not generated by ProofGen, is negligible.

We prove the theorem as a series of games with interleaved analysis. The restrictions of the games for the adversary are gradually tightened.

Game-0: Game-0 is the first game, the security game defined at the beginning of this chapter.

Game-1: Game-1 is the same as Game-0, with a slight difference. The challenger keeps a list that stores all signed file tags that have been responded to in the tag query. If the

adversary submits an effective tag τ_d but is not in the list signed by the challenger, the challenger outputs failure and aborts.

Analysis: if an adversary causes the challenger outputs failure with nonnegligible probability in Game-1, we can use the adversary to construct a forger to break the unforgeability of the signature scheme.

If the adversary does not cause failure in Game-1, its view is identical in Game-0 and Game-1. Through the description in Game-1, we know that the verification algorithm and extraction algorithm will get the parameters $name, n, d$ and u_1, u_2, \dots, u_t from the tag τ_d , and these values can only be generated by the challenger.

Therefore, if the adversary's success probability in Game-0 and Game-1 has a nonnegligible difference, we can construct a simulator to break the existence of the signature scheme by using the adversary.

Game-2: Game-2 is the same as Game-1, with a slight difference. The challenger keeps a list of tag queries and responses initiated by all adversaries. If the adversary submits proof that proves the verification algorithm successfully but σ is not equal to $\prod_{(i,V_i) \in Q} \sigma_i^{V_i}$, the challenger outputs failure and aborts.

Analysis: it is assumed that the failed replica file is divided into equal-length n blocks, expressed as $F_d = \{m_{d,i}\}$, $1 \leq d \leq t$, $1 \leq i \leq n$, and the corresponding parameters are $name, n, d$ and u_1, u_2, \dots, u_t . The tag set of blocks ψ is generated by TagGen. Suppose $Q = \{(i, V_i)\}$, $i \in I$ is the query that leads to failure, and the proof that responds to the adversary is $\mu'_1, \mu'_2, \dots, \mu'_t$ and σ' . Let the expected response generated by an honest prover be $\mu_1, \mu_2, \dots, \mu_t$ and σ , where $\mu_d = \sum_{(i,V_i) \in Q} V_i \cdot m_{d,i}$ and $\prod_{(i,V_i) \in Q} \sigma_i^{V_i}$. According to the proof of correctness, we know that the expected response satisfies the equation $e(\sigma, g) = e(\prod_{(i,V_i) \in Q} \prod_{d=1}^t H(name\|i)^{V_i} \cdot u_d^{\mu_d}, \nu)$. According to the description of Game-2, the adversary's response can also satisfy the equation $e(\sigma', g) = e(\prod_{(i,V_i) \in Q} \prod_{d=1}^t H(name\|i)^{V_i} \cdot u_d^{\mu'_d}, \nu)$, but $\sigma' \neq \sigma$. If there is $\mu'_d = \mu_d$ for each d , it satisfies the equation $\sigma' = \sigma$, which contradicts the above assumption. Therefore, let $\Delta\mu_d = \mu'_d - \mu_d$, and we know that at least one of $\{\Delta\mu_d\}$ is not 0.

Now we prove that if the adversary leads to the challenger outputs failure in Game-2 with nonnegligible probability, we can construct a simulator to solve the computational Diffie-Hellman problem.

The input value of the simulator is $g, g^{sk}, h \in G$, and its goal is to output h^{sk} . The behavior of the simulator is similar to the challenger in Game-1, but there are the following differences:

- (1) When generating the key, it sets the public key to g^{sk} received in the challenge, which means the simulator does not know the secret key sk .
- (2) The simulator programs the random oracle H and keeps a list of queries and responses. When the adversary randomly selects $r \xleftarrow{R} Z_p$ to query, its response is $g^r \in G$. It also responds to queries $H(name\|i)$ in a particular way, seen later.

- (3) When asked to store some file whose coded representation comprises the n blocks $\{m_{d,i}\}$, $1 \leq d \leq t$, $1 \leq i \leq n$, the simulator behaves as follows. It chooses a name $name \xleftarrow{R} Z_p$ at random. Because the space for choosing the name is large enough, the probability that the simulator chooses a name that has been queried by $name\|i$ in random oracle H is negligible.

For each d , $1 \leq d \leq t$, the simulator chooses random values $\delta_d, \gamma_d \xleftarrow{R} Z_p$ and set $u_d = g^{\delta_d} \cdot h^{\gamma_d}$. For each i , $1 \leq i \leq n$, the simulator chooses a random value $r_i \xleftarrow{R} Z_p$, and the response of the random oracle H is

$$H(name\|i) = \frac{g^{r_i}}{\left(g^{\sum_{d=1}^t \delta_d \cdot m_{d,i}} \cdot h^{\sum_{d=1}^t \gamma_d \cdot m_{d,i}}\right)}. \quad (2)$$

Now the simulator can calculate because we have

$$\begin{aligned} H(name\|i) \cdot \prod_{d=1}^t u_d^{m_{d,i}} &= \left(\frac{g^{r_i}}{\left(g^{\sum_{d=1}^t \delta_d \cdot m_{d,i}} \cdot h^{\sum_{d=1}^t \gamma_d \cdot m_{d,i}}\right)} \right) \cdot \prod_{d=1}^t u_d^{m_{d,i}} \\ &= \left(\frac{g^{r_i}}{\left(g^{\sum_{d=1}^t \delta_d \cdot m_{d,i}} \cdot h^{\sum_{d=1}^t \gamma_d \cdot m_{d,i}}\right)} \right) \\ &\quad \cdot \left(g^{\sum_{d=1}^t \delta_d \cdot m_{d,i}} \cdot h^{\sum_{d=1}^t \gamma_d \cdot m_{d,i}} \right) \\ &= g^{r_i}. \end{aligned} \quad (3)$$

Therefore

$$\begin{aligned} \sigma_i &= \prod_{d=1}^t \sigma_{d,i} = \prod_{d=1}^t \left(H(name\|i) \cdot u_d^{m_{d,i}} \right)^{sk} \\ &= \left(H(name\|i) \cdot \prod_{d=1}^t u_d^{m_{d,i}} \right)^{sk} \\ &= (g^{sk})^{r_i}. \end{aligned} \quad (4)$$

- (4) The simulator continues to interact with the adversary until the particular situation defined by Game-2 occurs: the adversary successfully proves the verification with σ' that is different from the expected σ .

The analysis of Game-0 and Game-1 ensures that the parameters $name, n, \{u_d\}, \{m_{d,i}\}, \{\sigma_i\}$ used in the protocol are generated by the challenger; otherwise, it will output

failure. This means that these parameters are generated gradually by the simulator described above. By dividing the tag σ' and the expected tag σ , we get

$$\begin{aligned} e\left(\frac{\sigma'}{\sigma}, g\right) &= e\left(\prod_{d=1}^t u_d^{\Delta\mu_d}, v\right) \\ &= e\left(\prod_{d=1}^t (g^{\delta_d} \cdot h^{\gamma_d})^{\Delta\mu_d}, v\right) \\ &= e\left(g^{\sum_{d=1}^t \delta_d \cdot \Delta\mu_d} \cdot h^{\sum_{d=1}^t \gamma_d \cdot \Delta\mu_d}, v\right). \end{aligned} \quad (5)$$

Rearranging terms yields

$$e\left(\sigma' \cdot \sigma^{-1} \cdot v^{-\sum_{d=1}^t \delta_d \cdot \Delta\mu_d}, g\right) = e\left(h^{\sum_{d=1}^t \delta_d \cdot \Delta\mu_d}, v\right). \quad (6)$$

Because $v = g^{sk}$, we find that the computational Diffie-Hellman problem has been solved:

$$h^{sk} = \left(\sigma' \cdot \sigma^{-1} \cdot v^{-\sum_{d=1}^t \delta_d \cdot \Delta\mu_d}\right)^{-\sum_{d=1}^t \delta_d \cdot \Delta\mu_d} \quad (7)$$

Unless the denominator is 0, we know that at least one of $\{\Delta\mu_d\}$ is not 0, so the probability of the denominator being 0 can be ignored.

Therefore, we prove that if there is a nonnegligible difference between the adversary's probability of success in Game-1 and Game-2, we can construct a simulator to solve the computational Diffie-Hellman problem, as required.

Game-3: Game-3 is the same as Game-2, with a slight difference. If the adversary submits proof that explains the verification successfully, but at least one μ_d is not equal to $\sum_{(i,V_i) \in Q} V_i \cdot m_{d,i}$, the challenger outputs failure and aborts.

Analysis: make some definitions like Game-2. We suppose that the failed replica file is divided into equal-length n blocks, expressed as $F_d = \{m_{d,i}\}$, $1 \leq d \leq t$, $1 \leq i \leq n$, and the corresponding parameters are $name, n, d$ and u_1, u_2, \dots, u_t . The tag set of blocks ψ is generated by TagGen. Suppose $Q = \{(i, V_i)\}$, $i \in I$ is the query that leads to failure, and the proof that responds to the adversary is $\mu'_1, \mu'_2, \dots, \mu'_t$ and σ' . Let the expected response generated by an honest prover be $\mu_1, \mu_2, \dots, \mu_t$ and σ , where $\mu_d = \sum_{(i,V_i) \in Q} V_i \cdot m_{d,i}$ and $\sigma = \prod_{(i,V_i) \in Q} \sigma_i^{V_i}$. Game-2 has ensured that we get $\sigma' = \sigma$; only $\{\mu'_d\}$ and $\{\mu_d\}$ can be different. Define $\Delta\mu_d = \mu'_d - \mu_d$, $1 \leq d \leq t$; then at least one of $\{\Delta\mu_d\}$ is not 0.

Now we prove that if the adversary leads to the challenger outputs failure in Game-3 with nonnegligible probability, we can build a simulator to solve the discrete logarithm problem.

The input value of the simulator is $g, h \in G$, and its goal is to output x such that $h = g^x$. The behavior of the simulator is similar to the challenger in Game-2, but there are the following differences.

- (1) When it is required to store a file whose coded representation comprises the n blocks $\{m_{d,i}\}$, $1 \leq d \leq t$, $1 \leq i \leq n$, the simulator behaves according to TagGen. For each d , $1 \leq d \leq t$, the

simulator chooses random values $\delta_d, \gamma_d \xleftarrow{R} Z_p$ and sets $u_d = g^{\delta_d} \cdot h^{\gamma_d}$.

- (2) The simulator continues to interact with the adversary until the particular situation defined by Game-3 occurs: the adversary successfully proves the verification with $\{\mu'_d\}$ different from the expected $\{\mu_d\}$.

According to the analysis of Game-1, we know that the parameters $name, n, \{u_d\}, \{m_{d,i}\}, \{\sigma_i\}$ used in the protocol are generated by the simulator. According to the analysis of Game-2, we know $\sigma' = \sigma$.

Construct the verification equation with $\{\mu_d\}$, respectively, with

$$\begin{aligned} &e\left(\prod_{(i,V_i) \in Q} \prod_{d=1}^t H(name \| i)^{V_i} \cdot u_d^{\mu_d}, v\right) \\ &= e(\sigma, g) \\ &= e(\sigma', g) \\ &= e\left(\prod_{(i,V_i) \in Q} \prod_{d=1}^t H(name \| i)^{V_i} \cdot u_d^{\mu_d}, v\right), \end{aligned} \quad (8)$$

concluding that

$$\prod_{d=1}^t u_d^{\mu_d} = \prod_{d=1}^t u_d^{\mu'_d}, \quad (9)$$

and therefore

$$\begin{aligned} 1 &= \prod_{d=1}^t u_d^{\Delta\mu_d} \\ &= \prod_{d=1}^t (g^{\delta_d} \cdot h^{\gamma_d})^{\Delta\mu_d} \\ &= g^{\sum_{d=1}^t \delta_d \cdot \Delta\mu_d} \cdot h^{\sum_{d=1}^t \gamma_d \cdot \Delta\mu_d}. \end{aligned} \quad (10)$$

We find that the discrete logarithm problem has been solved:

$$h^{sk} = \left(\sigma' \cdot \sigma^{-1} \cdot v^{\sum_{d=1}^t \delta_d \cdot \Delta\mu_d}\right)^{\sum_{d=1}^t \gamma_d \cdot \Delta\mu_d}. \quad (11)$$

Unless the denominator is 0. However, we know that at least one of $\{\Delta\mu_d\}$ is not 0, so the probability of the denominator being 0 can be ignored.

Therefore, we prove that if there is a nonnegligible difference between the adversary's probability of success in Game-2 and Game-3, we can construct a simulator to solve the discrete logarithm problem, as required.

4.1. Wrapping Up. Suppose the signature algorithm used to generate file tags is existential unforgeability, the computational Diffie-Hellman problem on bilinear groups and the discrete logarithm problem are difficult. In that case, there is a nonnegligible difference between the adversary's probability of success in Game-3 and Game-0. From Game-1 to

TABLE 4: Comparison of schemes' characteristics.

	[17]	[18]	[19]	[20]	[21]	[22]	Our scheme
Public verification	√	√	√	√	√	√	√
Batch verification	×	√	√	×	×	√	√
Privacy preserving	×	×	√	√	√	√	√
Fair payment	√	√	√	√	√	√	√
Interactivity	√	×	×	√	×	×	√
Multireplica	×	√	×	×	√	√	√

TABLE 5: Algorithm running time of our scheme ($N=64$ KB, $n=1024$, $c=300$, $t=3$).

Algorithm	ReplicaGen	TagGen	ProofGen	ProofVerify	Total
Time(s)	0.009413	3.376702	0.222226	0.003383	3.611731

Game-3, it is limited that τ_d, σ, μ_d , only the proof not correctly calculated by ProofGen, can respond to the challenge gradually, and Game-0 is the security game of our scheme. Therefore, the probability that an adversary who breaks through the security of our scheme uses the proof not generated by ProofGen to successfully prove the verification is negligible. This completes the proof of Theorem 1.

5. Performance Evaluation

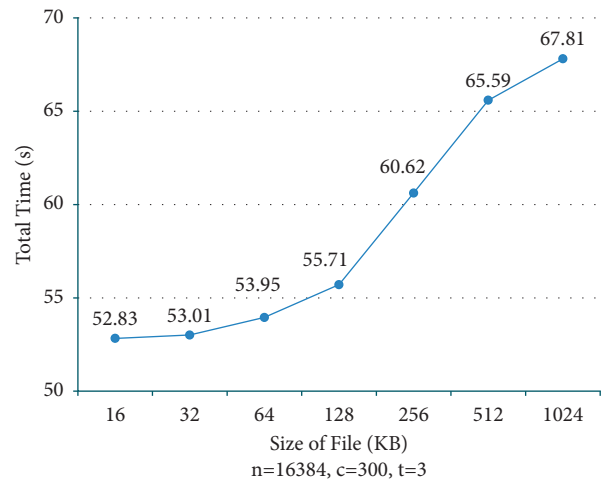
5.1. Comparative Analysis. We compare our scheme with other similar schemes from the dimensions of public verification, batch verification, privacy preserving, fair payment, interactivity, and multireplica. It can be seen from Table 4 that compared with other blockchain-based provable data possession schemes. Our scheme uses multireplica technology, based on BLS signature and homomorphic verification tag, combined with the smart contract with deposit mechanism, and it achieves all functions well.

5.2. Experiments. To evaluate the performance of our scheme, we designed a prototype of the scheme with C Programming language. For the large integer and pairing operations, we use the GMP Library (version 6.2.1) and PBC Library (version 0.5.14), respectively. AES256 is used for file encryption, and the length of signature key is 160 bits. The experimental environment is Intel Core i7 2.6 GHz, memory is 16 GB 2133mhz lpdrr3, and the operating system is macOS 12.0.1.

In the experiment, we mainly focus on the total running time of the scheme, including ReplicaGen time, TagGen time, ProofGen time, and ProofVerify time, as well as the relationship between the running time and file size N , number of blocks n , and number of challenge blocks c .

According to the study of [2], it assumes that the CSP destroys 1% of the data blocks uploaded by the DO; when $c=300$ and 460, the probability that DO can detect misbehavior of CSP is 95% and 99%.

The fixed file size is $N=64$ KB, the number of blocks is $n=1024$, the number of challenge blocks is $c=300$, and the number of replicas is $t=3$. The experimental results of the

FIGURE 4: The relationship between total time and file size ($n=16384$, $c=300$, $t=3$).

running time of each algorithm are shown in Table 5. It can be seen from Table 5 that time is mainly consumed in the TagGen algorithm, which is consistent with our assumption. In this algorithm, tags need to be calculated for each data block, and many exponential operations need to be carried out. Hence, the computational complexity is much higher than that in other algorithms.

Next, we fixed the number of blocks $n=16384$, the number of challenge blocks $c=300$, and the number of replicas $t=3$ and gradually increased the file size N from 16 KB to 1024 KB. The experimental results are shown in Figure 4. We can see that the relationship between the total time and the file size is basically linear, and the gap is not apparent when N is very small.

Then we fixed the file size $n=64$ KB, the number of challenge blocks $C=300$, gradually changed the number of blocks from $n=1024$ to 16384, and took the number of replicas t as 1, 2, and 3, respectively. The experimental results are shown in Figure 5. We can see that the total time increases exponentially with the number of blocks, and the total time increases with the number of replicas, but the impact is limited.

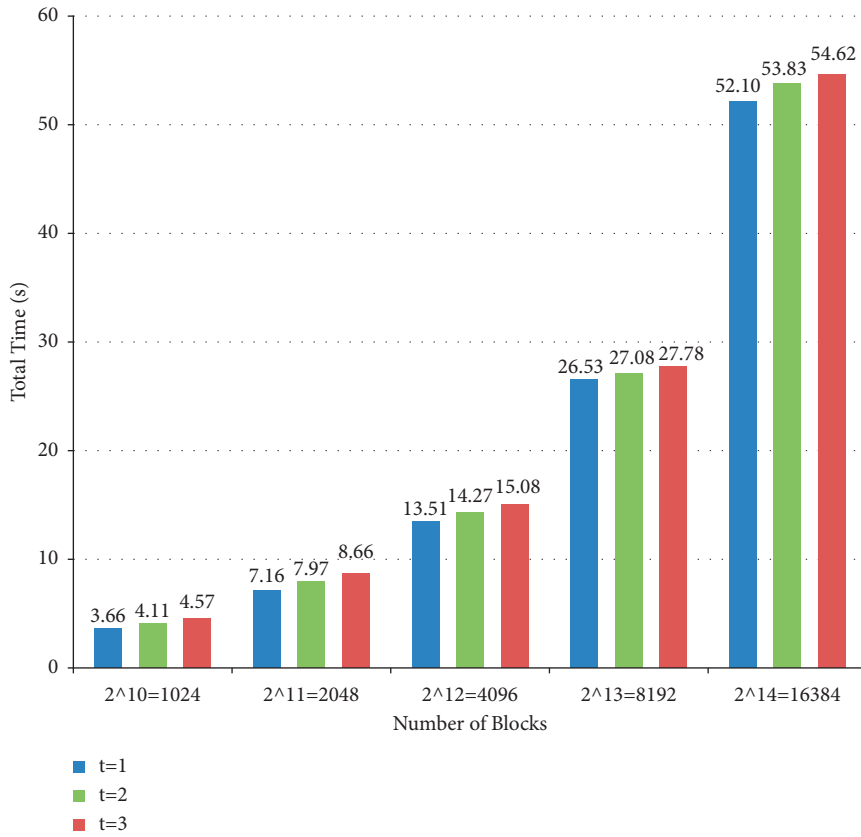


FIGURE 5: The relationship between total time and the number of blocks and replicas ($N = 64$ KB, $c = 300$).

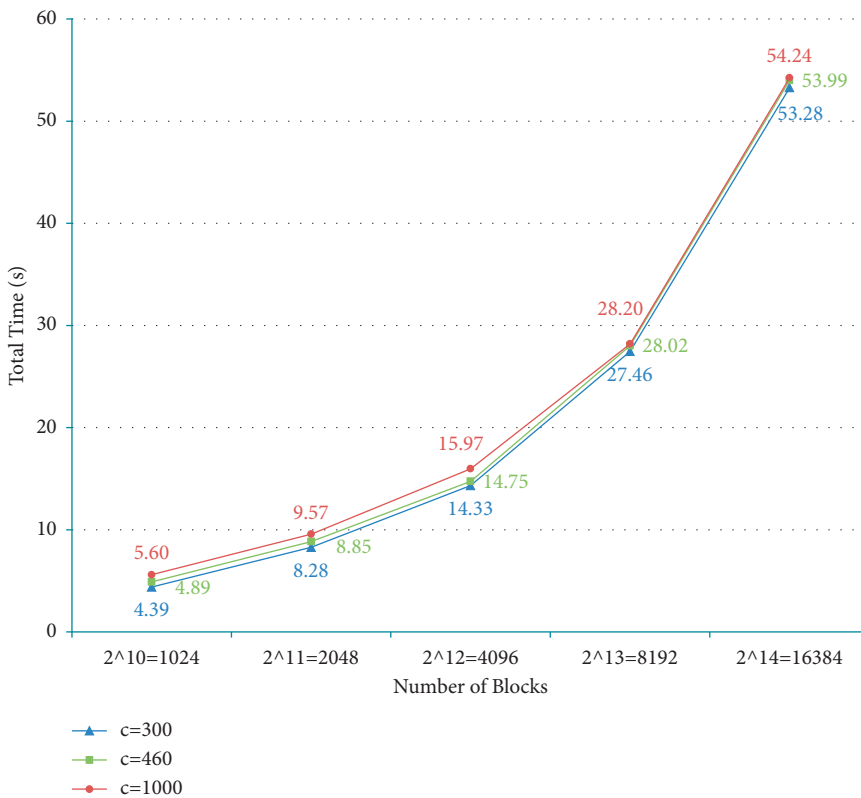


FIGURE 6: The relationship between total time and the number of blocks and challenge blocks ($N = 64$ KB, $t = 3$).

Finally, we fixed the file size $N = 64$ KB and the number of replicas $t = 3$, gradually changed the number of blocks from $n = 1024$ to 16384, and took the number of challenge blocks c as 300, 460, and 1000, respectively. The experimental results are shown in Figure 6. We can see that the total time increases with the number of challenge blocks, but the impact of the number of challenge blocks on the total time becomes smaller and smaller as the number of blocks increases.

6. Conclusion

This paper proposes a noninteractive multireplica provable data possession scheme based on smart contracts. The scheme replaces TPA with the smart contract, which provides a trusted environment for data integrity verification so that the verification process is public, tamper-proof, traceable, and periodically automatic. To reduce the transaction fees caused by the frequent operation of blockchain in the verification process, the concept of noninteractive is introduced. By presetting payment rules in smart contracts, fair transactions between the parties involved are guaranteed. The contracts will be executed automatically once the conditions are met.

Data Availability

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

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 Key R&D Program of China (nos. 2020YFB1708600 and 2020YFB1805403) and Foundation of Guizhou Provincial Key Laboratory of Public Big Data (nos. 2017BDKFJJ015, 2018BDKFJJ008, 2018BDKFJJ020, and 2018BDKFJJ021).

References

- [1] Seagate Rethink Data, *Put More of Your Business Data to Work—From Edge to Cloud*, Seagate Rethink Data, Chennai, India, 2020.
- [2] G. Ateniese, R. Burns, R. Curtmola et al., "Provable Data Possession at Untrusted Stores," in *Proceedings of the 14th ACM conference on Computer and communications security*, ACM, NY, USA, October 2007.
- [3] A. Juels and J. B. Kaliski, "Pors: Proofs of Retrievability for Large Files," in *Proceedings of the 14th ACM conference on Computer and communications security*, ACM, NY, USA, October 2007.
- [4] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 90–107, Melbourne, VIC, Australia, December 2008.
- [5] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," *Computer Security-ESORICS 2009*, vol. 5789, pp. 355–370, 2009.
- [6] C. Wang, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceedings of the IEEE INFOCOM*, San Diego, CA, USA, March 2010.
- [7] R. Curtmola, "Multiple-Replica Provable Data Possession," in *Proceedings of the 2008 the 28th International Conference on Distributed Computing Systems*, IEEE, Beijing, China, June 2008.
- [8] Z. Hao and N. Yu, "A Multiple-Replica Remote Data Possession Checking Protocol with Public Verifiability," in *Proceedings of the 2010 Second International Symposium on Data, Privacy, and E-Commerce*, IEEE, Buffalo, NY, USA, September 2010.
- [9] J. Wei, "Efficient dynamic replicated data possession checking in distributed cloud storage systems," *International Journal of Distributed Sensor Networks*, vol. 2016, Article ID 1894713, 2016.
- [10] Z. Ya-xing, "Multiuser and multiple-replica provable data possession scheme based on multi-branch authentication tree," *Journal on Communications*, vol. 36, no. 11, pp. 80–91, 2015.
- [11] S. Peng, F. Zhou, J. Li, Q. Wang, and Z. Xu, "Efficient, dynamic and identity-based remote data integrity checking for multiple replicas," *Journal of Network and Computer Applications*, vol. 134, pp. 72–88, 2019.
- [12] H. Yu, Z. Yang, M. Waqas et al., "Efficient dynamic multi-replica auditing for the cloud with geographic location," *Future Generation Computer Systems*, vol. 125, pp. 285–298, 2021.
- [13] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1204–1221, 2022.
- [14] T. Li, L. Chunmei, J. Yanling, and Y. Yixian, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, pp. 1–22, 2021.
- [15] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
- [16] Y. Chen, S. Dong, T. Li, Y. Wang, and H. Zhou, "Dynamic multi-key FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5239–5249, 2021.
- [17] T. Li, "Rational Protocols and Attacks in Blockchain System," *Security and communication networks*, vol. 2020, Article ID 8839047, 2020.
- [18] J. Xue, C. Xu, and L. Bai, "DStore: a distributed system for outsourced data storage and retrieval," *Future Generation Computer Systems*, vol. 99, pp. 106–114, 2019.
- [19] J. Xue, "Identity-based Public Auditing for Cloud Storage Systems against Malicious Auditors via Blockchain," *Science China-Information Sciences*, vol. 62, Article ID 0321043, 2019.
- [20] X. Yang, X. Pei, M. Wang, T. Li, and C. Wang, "Multireplica and multi-cloud data public audit scheme based on blockchain," *IEEE ACCESS*, vol. 8, pp. 144809–144822, 2020.
- [21] P. Huang, K. Fan, H. Yang, K. Zhang, H. Li, and Y. Yang, "A collaborative auditing blockchain for trustworthy data integrity in cloud storage system," *IEEE ACCESS*, vol. 8, pp. 94780–94794, 2020.

- [22] Y. Xu, "Blockchain empowered arbitrable data auditing scheme for network storage as a service," *IEEE TRANSACTIONS ON SERVICES COMPUTING*, vol. 13, no. 2, pp. 289–300, 2020.
- [23] D. Chen, H. Yuan, S. Hu, Q. Wang, and C. Wang, "BOSSA: a decentralized system for proofs of data retrievability and replication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 4, pp. 786–798, 2021.
- [24] K. Fan, Z. Bao, M. Liu, A. V. Vasilakos, and W. Shi, "Dredas: decentralized, reliable and efficient remote outsourced data auditing scheme with blockchain smart contract for industrial IoT," *Future Generation Computer Systems*, vol. 110, pp. 665–674, 2020.
- [25] H. Wang, H. Qin, M. Zhao, X. Wei, H. Shen, and W. Susilo, "Blockchain-based fair payment smart contract for public cloud storage auditing," *Information Sciences*, vol. 519, pp. 348–362, 2020.
- [26] Y. Li, Y. Yu, R. Chen, X. Du, and M. Guizani, "IntegrityChain: provable data possession for decentralized storage," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1205–1217, 2020.
- [27] R. Chen, Y. Li, Y. Yu, H. Li, X. Chen, and W. Susilo, "Blockchain-based dynamic provable data possession for smart cities," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4143–4154, 2020.
- [28] B. L. H. S. Dan Boneh, "Short Signatures from the Weil Pairing," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, Gold Coast, QLD, Australia, December 2001.
- [29] "Bitcoin: A peer-to-peer electronic cash system," 2009, <https://bitcoin.org/en/bitcoin-paper>.
- [30] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.
- [31] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," *Financial Cryptography and Data Security*, Springer International Publishing, vol. 10322, pp. 357–375, Cham, 2017.
- [32] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaría, "Blockchain and smart contracts for insurance: is the technology mature enough?" *Future Internet*, vol. 10, no. 2, p. 20, 2018.

Research Article

Blockchain Data Sharing Query Scheme based on Threshold Secret Sharing

Lu Chen ^{1,2,3} Xin Zhang ^{1,2,3} and Zhixin Sun ^{1,2,3}

¹Engineering Research Center of Post Big Data Technology and Application of Jiangsu Province, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

²Research and Development Center of Post Industry Technology of the State Posts Bureau (Internet of Things Technology), Nanjing University of Posts and Telecommunications, Nanjing 210003, China

³Engineering Research Center of Broadband Wireless Communication Technology of the Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Correspondence should be addressed to Zhixin Sun; sunzx@njupt.edu.cn

Received 11 December 2021; Accepted 1 March 2022; Published 6 April 2022

Academic Editor: Yuling Chen

Copyright © 2022 Lu 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.

Blockchain is a distributed ledger that combines technologies such as timestamp, cryptography, consensus mechanism, and peer-to-peer network. In the field of data recording and management, the blockchain data query scheme based on smart contracts consumes a lot of resources, and blockchain platforms that do not support smart contracts cannot achieve convenient data query. This study proposes a blockchain data sharing query scheme based on threshold secret sharing. The secret elements used to query data are shared through the Blakley space plane equation to limit the rights of the inquirer, ensuring the security of blockchain data query. At the same time, the Blakley space plane equation coefficient matrix is used to segment the data to be uploaded to the blockchain. It solves the problem that the data cannot be directly stored in the block due to their large size. It facilitates data uploading to the blockchain. The experimental results show that the additional time consumption of the secret sharing and recovery, data segmentation, and reconstruction of this scheme is much less than the block generation time. Therefore, this solution will not affect the normal operation of blockchain applications and can improve the security and the fault tolerance rate of data query.

1. Introduction

“Bitcoin: A Peer-to-Peer Electronic Cash System” [1], published in 2008, first introduced the concept of blockchain. This study describes the architectural concept of Bitcoin based on the peer-to-peer network [2], cryptographic mechanism [3], timestamp [4], and consensus mechanism [5]. At the same time, the concept of “chain of blocks” is introduced. The birth of Bitcoin has promoted the development of digital cryptocurrency. At this time, the focus of people’s attention is still on the “currency” attribute of Bitcoin, rather than the blockchain technology at the core of Bitcoin. After 2015, the emergence and subsequent rapid development of Ethereum [6] and Hyperledger [7] have changed this phenomenon. This made the underlying

blockchain technology more widely known and studied by more people. At present, the mainstream definition of blockchain in the industry is a combination of key technologies such as timestamps, cryptographic mechanisms, peer-to-peer transmission, and distributed consensus. It is a decentralized shared ledger with collective maintenance, immutability, security, and credibility [8].

According to different trust construction methods, blockchain can be divided into permissioned blockchain and nonpermissioned blockchain [9]. According to the different degrees of openness [10], the permission blockchain can be divided into consortium blockchain and private blockchain. The common application areas of blockchain can be divided into digital cryptocurrency, data recording and management, information security, and other fields [11].

Applications in the field of data recording and management include data storage, data authentication, information sharing, and copyright protection. Some researchers use blockchain for logistics data management. The blockchain-based logistics data security storage system stores logistics records in the blockchain to ensure that the entire logistics process can be audited. At the same time, the throughput of the blockchain-based IoT system is improved through the group-based POW mechanism [12]. Blockchain can also be used for public auditing of data, enabling the secure sharing of recorded information and the traceability of user identities [13]. Some researchers used the characteristics of blockchain technology to achieve efficient sharing of government information resources and digital rights management [14, 15]. The core of the application of blockchain in data recording and management scenarios is data, and these applications need to provide blockchain data query services, so the security of data queries is very important.

The smart contract is a computer program that can run premade rules on a distributed ledger. It can execute and verify the complex behavior of distributed nodes [16]. Smart contracts can be executed more effectively in a decentralized and trustless environment like the blockchain. At present, many blockchain applications have adopted smart contract-based methods to provide blockchain data query services. The query methods based on the smart contract deploy the data access rules directly into the blockchain smart contract after negotiation and approval. When the conditional parameters carried in the request meet the access rules, the smart contract will continue to execute, without relying on nonblockchain systems. However, the query of data in blockchain applications is often only associated with a specific node. The smart contract-based approach requires each node to run the deployed contract, and the resource consumption of blockchain applications is relatively large.

At the same time, in the digital encryption currency field, the data size that the block body of the early blockchain platform can accommodate does not match the needs of nondigital encryption currency scenarios. Therefore, some blockchain-based data applications can only store relevant data indexes on the blockchain, and the original complete data need to be stored on the off-chain storage system. Literature [17] proposed a medical consortium blockchain system based on the PBFT consensus mechanism. The consortium blockchain in the system is responsible for storing the ID of the transaction order corresponding to the traditional database table, and the specific transaction order and original medical data still need to be saved to the existing medical information system. In the electronic medical record sharing model [18], the desensitized electronic medical record will be stored off-chain, and the off-chain index of the desensitized medical record will become the leaf node of the Merkle tree in the block.

Therefore, in view of the blockchain data query requirements in the data recording and management scenario, this study builds a secure and reliable blockchain data sharing query scheme based on the threshold secret sharing mechanism and erasure codes. The main contributions of this study are as follows:

- (1) A blockchain data sharing query scheme based on the Blakley threshold secret sharing mechanism is proposed. The secret elements used to query data are shared through the Blakley space plane equation, limiting the rights of blockchain data inquirers, and it further ensures the security of blockchain data query.
- (2) An erasure code-based data uploading method is proposed. The coefficient matrix of the Blakley space plane equation is used as the erasure code encoding matrix to segment the data. This method not only solves the problem that the data cannot be directly stored in the block because of the large size of the block but also improves the fault tolerance rate during query.
- (3) Simulation experiments and analysis of the scheme are carried out. The results show that the rate of secret sharing and recovery, data segmentation, and reconstruction of this scheme is higher than that of block generation. This scheme does not affect the normal operation of the blockchain system; at the same time, it improves the security of data query.

2. Related Work

Many researchers design blockchain data query methods based on smart contracts. In the blockchain-based digital archive protection and sharing system proposed in reference [19], the consortium blockchain manages digital archives and users, the public blockchain regularly stores block snapshots of consortium blockchain, and the private IPFS stores ciphertext digital archives. The management tasks of the members of the internal consortium blockchain are undertaken by the digital identity management contract. Literature [20] designed a medical data sharing platform based on blockchain. When the access statement sent by the data requester matches the consent statement specified by the data provider, the smart contract will continue to execute the medical data sharing process. In the medical privacy data sharing query model proposed in reference [21], the access list contract is deployed by the medical institution to frame the private data collection belonging to the medical research category. At the same time, users also have the right to remove institutions from their own data authorized access list. Literature [22] constructed a paid data query scheme based on smart contracts, which built data access clauses into smart contracts along with data hashes. Only users who meet the access terms can get the key in the contract and the data hash in the trusted environment and finally retrieve the data.

The above research all use smart contracts for data query, which requires each node of the blockchain to run deployed smart contracts. Therefore, the data query methods based on smart contracts will make the resource consumption of blockchain applications larger.

In addition to smart contract-based methods, many scholars at home and abroad currently adopt blockchain data query schemes based on cryptographic mechanisms. Huang et al. [23] stored the access strategy of attribute-based

encryption (ABE) and user attributes in the database. The metadata were uploaded to the blockchain system, and the blockchain was used to ensure data security. The solution uses attribute-based encryption to control the access and query of blockchain data. In order to improve the security of data sharing between enterprises, Wang et al. [24] proposed a shared query and access model based on ABE and blockchain, which is composed of two blockchains responsible for the internal and interenterprise. The model uses attribute-based encryption to protect internal data access and data sharing between enterprises. Thwin and Vasupongayya [25] designed a medical data sharing query scheme based on proxy reencryption. The medical ciphertext data were uploaded to the chain through the gateway server after reencryption. The cloud storage service saves medical data, and the private chain stores medical metadata and raw data access logs to improve the confidentiality and security of personal medical data. Truong et al. [26] adopted prefix encryption technology for hierarchical structure scenarios to improve the security of data queries between IoT devices. In this scheme, the data requester first uses the ciphertext data encrypted by the prefix to apply to the data provider. After obtaining the consent of the data provider, the requester obtains the key sent by the data provider from the key authority and decrypts the ciphertext.

The above research all use cryptographic mechanisms to control access rights and protect the query of blockchain data. But due to the characteristics of the blockchain, it is determined that the blockchain data need to be shared and inquired more. This requires the data owner to know the specific inquirers in advance to implement effective authority control. Therefore, the solutions in the above research cannot quickly respond to the inconsistency between the actual inquirers and the preset inquirers. So, they are not flexible enough.

The query method based on threshold secret sharing can determine the approximate query permission range in advance. So, it is suitable for sharing query scenarios and is more flexible. At the same time, only when the number of people applying for reconstruction exceeds the preset threshold, the data can be successfully queried, which limits the rights of the inquirer and improves security. In the medical privacy data sharing scheme based on the consortium chain [27], medical institutions distribute key shares through the Shamir secret sharing mechanism to medical users who propose to share patient medical privacy data. The system then restores the key through enough key shares uploaded by medical users. Finally, medical users can query the private data. Xu et al. [28] proposed a verifiable secret sharing and multi-party coordination mechanism that can reduce the complexity and cost of interaction in a blockchain-based cyber-physical system. The user node sends an application to the witness node responsible for verifying the $1/n$ core shares distributed by the sensor node. Only if it obtains valid responses from more than n witness nodes, it can obtain permission to view the data. Sohrabi et al. [29] introduced a master node responsible for storing user key shares distributed by smart contracts. After the nodes that meet the query conditions successfully obtain the ciphertext data, they need to continue to send the key share request. The

user can view the data only after receiving a certain valid key share response. Literature [28, 29] distributed the key to multiple places through a secret sharing mechanism, but only one data query request can be served. When the number of people applying for query is n , the system needs to transmit the key share set n times. So, these methods have a large amount of transmission. Table 1 summarizes the features of existing blockchain data query schemes.

Therefore, this study proposes a blockchain data sharing query scheme based on the Blakley threshold secret sharing mechanism. This scheme adopts the mode of multiple key shares and multiple query users to reduce transmission costs. It improves the security of data queries while improving the fault tolerance of the system through the erasure code mechanism.

3. Preliminary Knowledge

3.1. Threshold Secret Sharing. Given positive integers t and $n(n \geq t)$, the secret information S is divided into n subsecrets S_i and they are distributed to n different people, where $1 \leq i \leq n$. These people are respectively denoted as $\{p_1, p_2, \dots, p_n\}$. When the number of subsecrets is greater than or equal to t , the secret S can be reconstructed; conversely, when the number of subsecrets is less than t , S cannot be reconstructed. At this time, the positive integer t becomes the threshold value, and the above process is called (t, n) threshold secret sharing.

Shamir proposed a threshold scheme based on algebraic Lagrange interpolation polynomials in 1979, namely, Shamir secret sharing [30]. First, the prime number q is determined, the finite field F_q is selected, and the secret information S is made to satisfy $S \in F_q$. Then, different nonzero elements x_1, x_2, \dots, x_n are chosen on F_q . Then, $t-1$ elements a_1, a_2, \dots, a_{t-1} are chosen on F_q , and a polynomial $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1}$ is constructed, where a_0 is the secret S . $f(x_i)$ is obtained through the polynomial, where $1 \leq i \leq n$. Then, $(x_i, f(x_i))$ is sent to the corresponding p_i . When there are t people trying to reconstruct the secret S , they need to solve the following equation:

$$f(x) = \sum_{i=1}^t f(x_i) \prod_{1 \leq j \leq i, j \neq i} \frac{x - x_j}{x_i - x_j} \text{ mod } q. \quad (1)$$

Then, $S = f(0)$ can be calculated.

Blakley constructed another secret sharing scheme through spatial geometry [31]. It regards the secret S as a point in the t -dimensional space and divides S into n $(t-1)$ -dimensional linearly independent spaces. Then, any t linearly independent $t-1$ dimensional spaces can determine a unique point, which is the secret S . Therefore, only $t \times n + n$ elements need to be selected in the finite field F_q to form a t -element linear equation set:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1t}x_t = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2t}x_t = b_2, \\ \dots, \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nt}x_t = b_n. \end{cases} \quad (2)$$

TABLE 1: Feature comparison of blockchain data query schemes.

	Type of blockchain	Application scenarios	Data query technology adopted	Advantages	Disadvantages
Literature [19]	Consortium blockchain, public blockchain	Digital archive management	Smart contract		
Literature [20]	Consortium blockchain	Medical data sharing	Smart contract	Queries are automatically executed when user requests meet preset rules and do not rely on nonblockchain systems.	Each node needs to run the deployed contract, and the resource consumption of blockchain applications is large.
Literature [21]	Consortium blockchain	Medical privacy data sharing	Smart contract		
Literature [22]	Public blockchain	Paid query of data	Smart contract		
Literature [23]	Public blockchain	Data security sharing	Attribute-based encryption		
Literature [24]	Private blockchain, consortium blockchain	Enterprise data sharing	Attribute-based encryption	Easy to add an access permission control mechanism, which makes access permission control more fine-grained.	Inability to quickly respond to situations where the actual inquirers do not match the preset inquirers.
Literature [25]	Private blockchain	Medical data sharing	Proxy reencryption		
Literature [26]	Consortium blockchain	IoT device data query	Prefix encryption		
Literature [27]	Consortium blockchain	Medical privacy data sharing	Secret sharing	Determine the general query permission scope in advance, which is suitable for shared query scenarios. Data can be queried only when the number of users applying for reconstruction exceeds the preset threshold.	There is no data reconstruction mechanism. Unable to recover data when there is a failed node. Only one data query request can be served. When the number of inquiries is large, the system transmission volume is large.
Literature [28]	Not specified	Cyber-physical system	Secret sharing		
Literature [29]	Public blockchain	Cloud data sharing	Secret sharing		

The above equations can be denoted as $AX = B$. In the coefficient matrix A , any t rows are required to be linearly independent, and the secret S must be the only solution of the equation set. Therefore, when there are t people trying to recover the secret S , they only need to solve the equation set containing these t linearly independent lines to solve the secret S .

3.2. Erasure Codes. Erasure codes [32] originated in the field of communications. They are used to correct errors in data transmission. In the field of data storage, the original data D can be divided into t data blocks. Then, they are coded with erasure codes, to obtain a total of n data blocks and redundant blocks. Any t redundant blocks can recover the original data D according to the nature of erasure codes.

Reed-Solomon code [33] (Reed-Solomon, referred to as RS code) is a horizontal encoding that performs polynomial operations on the elements of the Galois field $GF(2^w)$. Common RS coding matrices are Vandermonde matrix and Cauchy matrix. The construction of Vandermonde matrix [33] is relatively simple. In Galois field $GF(2^w)$, Vandermonde matrix undergoes elementary transformation, and the first t rows are transformed into the identity matrix E . However, although the addition on the Galois field $GF(2^w)$ can be converted into an exclusive OR, the multiplication needs to rely on the discrete logarithm operation. Therefore, the RS code based on the Vandermonde matrix has a large amount of calculation. The Cauchy matrix [34] can convert

the multiplication in the Galois field $GF(2^w)$ to the $GF(2)$ binary operation and turn the matrix multiplication into a simple binary XOR operation.

4. System Scheme

4.1. System Model. The system model of blockchain data sharing query scheme based on threshold secret sharing mainly involves three entities, namely, data user, shared user, and blockchain storage system.

Data User. The owner of the data to be uploaded. The data will be uploaded to the blockchain storage system after being encrypted and segmented.

Shared User. In order to ensure that the data uploaded by the user who owns data are complete and correct, a certain number of shared users are required to work together to share data.

Blockchain Storage System. A storage system that stores data uploaded by data users. If a data query request is received, the blockchain storage system will return the corresponding data result.

The system model of this scheme is shown in Figure 1. First, data users will need to upload data to the blockchain storage system for encryption operations. Next, data segmentation and erasure code encoding are performed on the encrypted data. Then, the data encoded by the erasure code are uploaded to the blockchain storage system to complete the storage of the data.

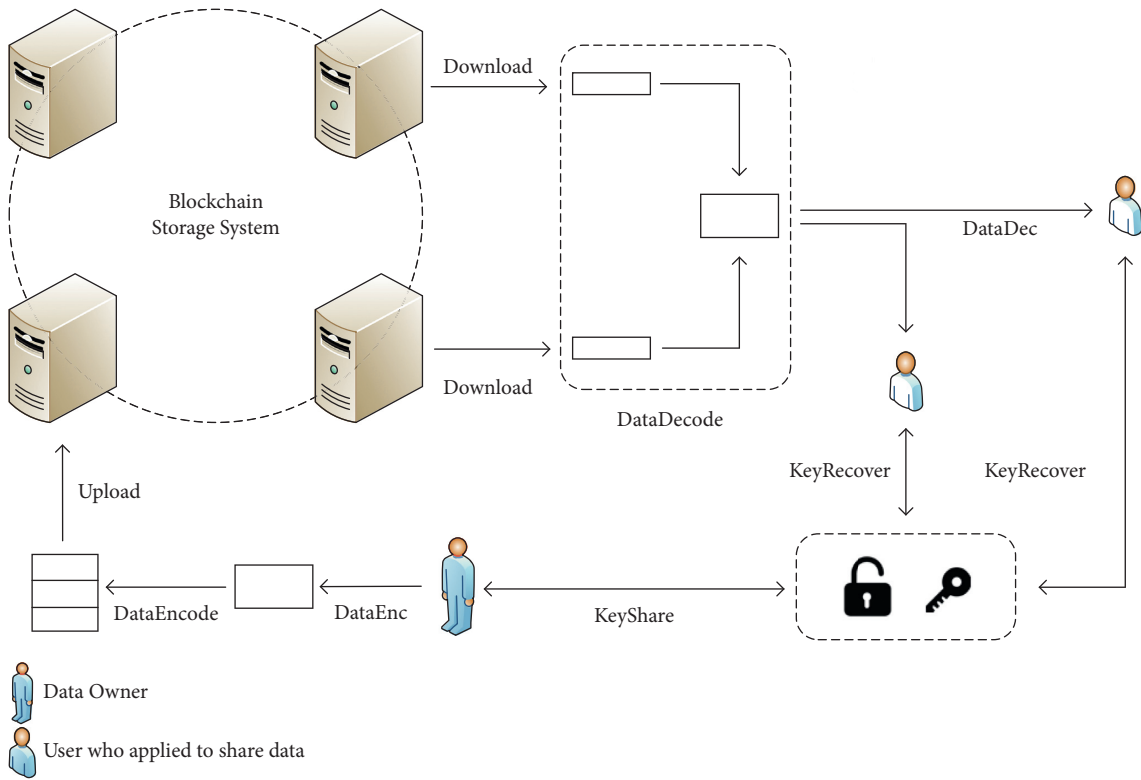


FIGURE 1: System model.

In order to ensure the consistency of data, shared users need to cooperate with each other to obtain and share data. That is, shared users need to apply to data users first. After the data user receives the application, the key is divided into multiple key shares through the Blakley secret sharing technology and then distributed to the shared user who requests. Shared users obtain the minimum required number of data coding blocks from the blockchain storage system, and then, the shared users work together to complete the reconstruction of the coding block. The key is recovered based on the multiple key shares held by shared users. Then, they complete the decryption of the data and obtain the original data.

The formal definition of this scheme is as follows:

- (1) $\text{KeyGen}(\lambda)$: Enter the security parameter λ , the algorithm output key K , and the key K is used to encrypt the original data;
- (2) $\text{DataEnc}(K, D)$: Input the key K , the original data D , and the algorithm outputs the encrypted ciphertext ED ;
- (3) $\text{MatrixGen}(\text{ID}, n, t)$: Enter the data user ID, and the algorithm uses different data user ID to determine different matrix factors. Then, generate different $n \times t$ -order nonsingular matrices M ;
- (4) $\text{DataEncode}(M, ED)$: Input matrix M , ciphertext data ED , and output code block $C = \cup_{i=0}^{n-1} C_i$ after encoding;
- (5) $\text{KeyShare}(K, M, n, t)$: Input the key K , matrix M , the number of shared users n , the threshold value t , and output the corresponding key share equation x ;

- (6) $\text{KeyRecover}(x)$: Input the key share equation x and output the key K ;
- (7) $\text{DataDecode}(M, C)$: Input coding matrix M , ciphertext coding data C , and output ciphertext data ED ;
- (8) $\text{DataDec}(K, ED)$: Enter the key K , the ciphertext data ED , and decrypt the original data D .

The security requirements of this solution are as follows:

- (1) When the shared users do not provide the geometric space equations that meet the preset number requirements, then they cannot recover the key correctly.
- (2) When the shared user obtains the spatial geometric equation used to recover the secret, the equation can be used multiple times to recover the secret.

4.2. Scheme Construction

4.2.1. Data Upload and Storage Stage. First, the data user executes the KeyGen algorithm to generate the key K that will be used later. Then, using the DataEnc algorithm, the original data D to be uploaded to the blockchain storage system is encrypted with the key K generated in the previous step. Then, data user uses the MatrixGen algorithm to generate a nonsingular matrix according to ID. The matrix will be used as an erasure code encoding matrix to encode the encrypted ciphertext data and the Blakley secret sharing

space geometric equation coefficient matrix. The data user uses the erasure code encoding matrix M as the input of the algorithm, DataEncode to encode and slice the ciphertext data, and ED to obtain the slice $\cup_{i=0}^{n-1} C_i$. Finally, the encoded block is sent to the blockchain storage system. The blockchain system stores the encoded ciphertext data C . The implementation of the data upload algorithm is shown in Figure 2.

4.2.2. Key Distribution Stage. Data users need to distribute keys to shared users. Both the Cauchy matrix and the Vandermonde matrix are nonsingular matrices. But the Cauchy matrix has a smaller computational time complexity than the Vandermonde matrix. The data user executes the MatrixGen algorithm to generate the Cauchy matrix M , and the matrix M is used as the coefficient matrix of the t -dimensional space geometric equation. The coefficient matrix M and a certain point K are used in the t -dimensional space to determine the unique t -dimensional space plane equation. Assuming that the current secret is shared with n users in total, then t of the n shared users can recover the secret K . The implementation of the shared user secret sharing algorithm is shown in Figure 3. I_i is the coefficient of the $t - 1$ -dimensional space equation, that is, the subsecret.

4.2.3. Key Recovery and Data Access Stage. In the key recovery and data access stage, the shared user first issues an access application and obtains the encoded ciphertext data from the blockchain storage system. However, the data are encrypted and encoded. So, the details of the data cannot be viewed directly. At this point, it is a binary string without any semantics for the viewer.

When many shared users trying to recover the ciphertext-encoded data C , they need to cooperate to recover the t -dimensional space plane equation set ($AK - B = 0$) through each user's own $t - 1$ -dimensional space plane equation. When the number of shared users reaches the number t presented by the data uploading user, the rank of the coefficient matrix of the aforementioned spatial plane equation system is equal to t , and the coefficient matrix can be reversed. This scheme uses the same matrix as the spatial plane equation coefficient matrix and the erasure code encoding matrix. At this time, the shared users can cooperate with each other to decode the ciphertext-coded data C through the inverse matrix. So, they obtain the ciphertext data ED. Then, the space plane equation of rank t is used to calculate the certain point K in the t -dimensional space, and this point is the key. Then, the original data D can be decrypted by the key. The shared user decoding and decryption algorithms are shown in Figure 4.

5. Experiment and Analysis

Based on Hyperledger Fabric1.0 and the erasure code processing library reedsolomon of Golang language, this section implements a blockchain-based threshold secret sharing query scheme on a server with Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50 GHz CPU and 32G memory.

Algorithm 1 Algorithm for Data Upload and Storage

```

Input:  $\lambda, D, n, t, ID$ 
Output: null
1: function UPLOADANDSTOREDATA ( $\lambda, D, n, t, ID$ )
2:    $K \leftarrow \text{KeyGen}(\lambda)$ 
3:    $ED \leftarrow \text{DataEnc}(K, D)$ 
4:   split data  $ED$  into  $ED_0, ED_1, \dots, ED_{t-1}$  linearly
5:    $M \leftarrow \text{MatrixGen}(ID, n, t)$ , let  $a_{i,j}$  as the element of Matrix  $M$ 
6:   for  $i = 0 \rightarrow n - 1$  do
7:     init  $C_i = 0$ 
8:     for  $j = 0 \rightarrow t - 1$  do
9:        $C_i += a_{i,j} \times ED_j$ 
10:    end for
11:  end for
12:  let  $C$  as  $\{C_0, C_1, \dots, C_{n-1}\}$ 
13:  send  $C$  to Blockchain Storage System
14:  StoreDataInBlockStorageSystem( $C$ )
15: end function

```

FIGURE 2: Data upload and storage process.

Algorithm 2 Algorithm for Key Distribution

```

Input:  $ID, n, t, K$ 
Output:  $I_0, I_1, \dots, I_{n-1}$ 
1: function DISTRIBUTEKEY ( $ID, n, t, K$ )
2:    $M \leftarrow \text{MatrixGen}(ID, n, t)$ , let  $a_{i,j}$  as the element of Matrix  $M$ 
3:   let  $K = \{k_0, k_1, \dots, k_{t-1}\}$ 
4:   for  $i = 0 \rightarrow (n - 1)$  do
5:     init  $B_i = 0, I_i = []$ 
6:     for  $j = 0 \rightarrow (t - 1)$  do
7:        $B_i += a_{i,j} \times k_j$ 
8:        $I_i.append(a_{i,j})$ 
9:     end for
10:     $I_i.append(B_i)$ 
11:  end for
12:  return  $I_0, I_1, \dots, I_{n-1}$ 
13: end function

```

FIGURE 3: Key distribution process.

5.1. Correctness Analysis. This section will analyze the correctness of shared users' key recovery in this scheme. The data user that needs to upload to the blockchain storage system is assumed D and the private key is assumed K . Then, the plaintext is encrypted to get $ED = \text{DataEnc}(K, D)$. According to the user's globally unique ID, the corresponding matrix factor is generated, and then, the $n \times t$ -th order nonsingular matrix M is generated: $M = \text{MatrixGen}(ID, n, t)$. The DataEncode(E, MD) algorithm is executed on the ciphertext data ED, and the $C = \cup_{i=0}^{n-1} C_i$ is uploaded to the blockchain storage system. When a sharing user applies for secret sharing, the data user constructs $n - t + 1$ -dimensional space plane equations through the matrix M and the key K . The sharer can determine the unique intersection point K through any $t - 1$ -dimensional space planes. There are two possibilities for the number of shared users n_s and t who applied for viewing:

Algorithm 3 Algorithm for Key Recovery and Data Access

```

Input:  $I_0, I_1, \dots, I_{k-1}$ 
Output:  $D$ 
1: function RECOVERKEYANDACCESSDATA ( $I_0, I_1, \dots, I_{k-1}$ )
2:   download the encoded ciphertext  $C$  from Blockchain Storage System
3:   init  $A = [ ]$ ,  $B = [ ]$ ;
4:   for  $i = 0 \rightarrow (k-1)$  do
5:     for  $j = 0 \rightarrow (t-1)$  do
6:        $a_{i,j} = I_{i,j}$ 
7:     end for
8:      $b_i = I_{i,t}$ 
9:   end for
10:   $r = \text{rank}(A)$ 
11:  if  $r \geq t$  then
12:    let  $p_{i,j}$  as the element of Matrix  $A^{-1}$ ;
13:    for  $i = 0 \rightarrow (t-1)$  do
14:      for  $j = 0 \rightarrow (n-1)$  do
15:         $ED_i = p_{i,j} \times C_j$ 
16:         $K_i = p_{i,j} \times B_j$ 
17:      end for
18:    end for
19:     $D = \text{DataDec}(K, ED)$ 
20:    return  $D$ 
21:  end if
22:  return null
23: end function

```

FIGURE 4: Key recovery and data access process.

- (1) $t \leq n_s \leq n$. At this time, the number of users applying for shared secrets n_s is greater than the minimum threshold t and less than the initially set number of secret shares n . Although at this time $n_s \leq n$, but in the coefficient matrix, any t rows in n rows are linearly independent. From the knowledge of linear algebra, there are t rows in any n_s rows of the coefficient matrix that are linearly independent. That is, among the n_s shared users, any t shared users can use their own space plane equations to determine the unique intersection point K .
- (2) $n_s > n$. If n_s shared users ($n_s > n$) share n $t-1$ -dimensional space plane equations (a $t-1$ -dimensional space plane equation will be held by multiple shared users), there is no guarantee that the shared users will hold exactly t linearly independent spatial plane equations when trying to recover the key. Therefore, all shared users are divided into $k = \lceil n_s/n \rceil$ groups. The data distribute n space plane equation coefficient matrices to each group of shared users. There are n shared users in each group. Only t linearly independent $t-1$ -dimensional space plane

equations can recover the secret. At this time, there are still $n_s - k \cdot n$ shared users who have not received the shared secret share. The data user first sends the $n_s - k \cdot n$ spatial plane equation coefficient matrices to $n_s - k \cdot n$ shared users. If $n_s - k \cdot n \geq t$, $n_s - k \cdot n$ shared users can directly recover the keys; if $n_s - k \cdot n < t$, then the data user can hold the remaining $t - (n_s - k \cdot n)$ space plane equations as a virtual shared user.

5.2. Security Analysis. In this scheme, the shared users restore the original data through erasure code decoding and Blakley secret sharing. For Blakley secret sharing, only when the number of shared users with the correct secret share exceeds the minimum threshold t , the unique solution of the full-rank t -ary linear equation system can be solved, and any number of users less than t cannot get the unique intersection point correctly. For the $n \times t$ -order erasure code matrix, when $(n-t)$ pieces of data are lost or tampered with, the original data can still be recovered through the erasure code decoding mechanism. When the number of lost or tampered data fragments exceeds $(n-t+1)$, it cannot be recovered. This scheme is a blockchain application for data recording and management scenarios. The open, transparent, and nontamperable characteristics of the blockchain make this probability almost nonexistent. This can only happen when the adversary has more than 51% of the computing power, but the cost of this is far greater than the value of the information itself.

5.3. Efficiency Analysis

5.3.1. Successful Reconstruction Rate. This scheme uses erasure code as the data slicing method. When shared users need to reconstruct data, the corresponding code package can be obtained from the blockchain storage system. The current blockchain storage system is assumed to have a total of p nodes. The storage system does not adopt a full copy redundancy scheme due to the limitation of node storage capacity, and each node only stores part of the data locally. When the data user uploads the data for storage, the erasure coding matrix is an $n \times t$ -order Cauchy matrix; that is, the ciphertext data ED are first divided into t data slices and then encoded and converted into n coded slices. The average storage capacity of each node in the current system is assumed to be only q ($q \leq n$) and the performance of each node is evenly distributed in the system. Then, the probability that the data can be directly restored through the local storage of the blockchain storage system node is as follows:

$$P_1 = 1 - \frac{C_n^{t-1}F(t-1, p, q) + C_n^{t-2}F(t-2, p, q) + \dots + C_n^qF(q, p, q)}{(C_n^q)^p} \quad (3)$$

Among them, $F(\text{num}, p, q)$ indicates that the current p nodes (the storage limit of each node is q) can completely store the number of combinations of num different blocks.

$$P_2 = 1 - \frac{C_t^{t-1}F(t-1, p, q) + C_t^{t-2}F(t-2, p, q) + \dots + C_t^q F(q, p, q)}{(C_t^q)^p} \quad (4)$$

When $n \geq t$, $C_n^x > C_t^x$, $0 < x < t$, $P_1 > P_2$ can be obtained. Therefore, the appropriate erasure code matrix size and node parameters can provide a high error tolerance rate.

5.3.2. Processing Rate. In this section, we will analyze the correctness of the shared user recovery key in this scheme. The three common key lengths are chosen as follows: 128 bit, 256 bit, and 512 bit. The current secret threshold is assumed to be $t = 10$, and the number of the shared people is set to 11 to 25. The simulation results are shown in Figure 5. When the secret length is 128 bit and the number of shared users is 25 (that is, any 10 users out of 25 users can recover the secret), the time spent on key distribution and recovery is 9.93 ms and 2.53 ms, respectively. When the secret length is 256 bit and 512 bit, the corresponding key distribution time is 10 ms and 20.73 ms, respectively. When the length of the secret is longer and the number of people sharing the secret is larger, the distribution time of the secret is also longer, but it remains at the millisecond level.

Before the data owner uploads the data and before the data inquirer views the data, the data need to be segmented and reconstructed. Therefore, the performance of the erasure code-based data sharing mechanism in our solution is related to the execution performance of the blockchain application. The original data size is set to 100 M, $t = 10$ (that is, the original ciphertext data is divided into 10 parts). Now, the effect of the number of different codes n on the encoding and decoding processing time is tested. The number of code blocks is set to 11–20. When $n = 11$, the processing time was the shortest, encoding and decoding took 225.8 ms and 156.6 ms, respectively, and the encoding and decoding rates were 442.86 MB/s and 638.57 MB/s, respectively. As shown in Figure 6, the larger the number of encoding blocks, the longer the encoding and decoding time. As the number of redundant data fragments increases (that is, the matrix size continues to increase), the encoding and decoding rate gradually decreases, as shown in Figure 7.

Therefore, our solution uses the Blakley space plane equation coefficient matrix to perform erasure-coded data fragmentation processing, which reduces the data size, and this does not significantly affect the speed and performance of blockchain applications in data recording and management scenarios.

5.4. Adaption Analysis. This section will analyze the adaptability between our scheme and the blockchain system. As shown in Figure 8, the relationship between our scheme and

If the data uploaded by the data user are sliced and uploaded directly without encoding, then the probability that the data can be directly recovered from the blockchain storage system is as follows:

blockchain architecture [8] mainly includes the data layer and network layer. In our scheme, coding algorithm, data segmentation, and reconstruction modules are added to the data layer. At the same time, the data divided into t slices will generate n slices after erasure coding. There are redundant $(n - t)$ data fragments. This mechanism can increase the storage cost of a piece of data by (n/t) times.

At the network layer, the scheme adds computing and network resources of data users, shared users, and nodes of blockchain storage system. This improves the correct reconstruction rate of data to be stored in the system and the security of data sharing. It is assumed that the ciphertext data ED is encoded by the $n \times t$ -order erasure matrix and uploaded to the blockchain storage system. That is, the ciphertext data are segmented into t slices, and the amount of data in each slice is d . Then, the amount of data transferred is shown in the following formula:

$$O_t = k \cdot d, \quad t \leq k \leq n. \quad (5)$$

As shown in the blue area in Figure 9, the amount of data O_t to be transmitted is between $n d$ and $t d$, and the shared user can successfully recover the original data. At the same time, the number of verification times is linearly related to the number of fragments when the data fragments are cut and encoded.

According to the above content, the improvements made in this scheme increase the cost of the transmission and verification mechanism at the network layer, but do not change their internal operation mechanism. Similarly, if the block structure is not modified, the erasure code mechanism introduced will only increase the block storage cost without affecting the existing block storage structure. According to the analysis in Section 5.3, our scheme can provide a high reconstruction success rate without affecting the speed and performance of blockchain applications in data recording and management scenarios. Changes in the data layer and network layer did not break the original mechanism but only increased the cost. Therefore, our solution does not affect the decentralized, sequential data, collective maintenance, programmability, and security and trusts features of blockchain. Therefore, our scheme is completely suitable for the blockchain storage system.

5.5. Comparison. As summarized in Table 2, we compare and analyze our scheme and the other two blockchain data query schemes. The query mode of literature [28] and

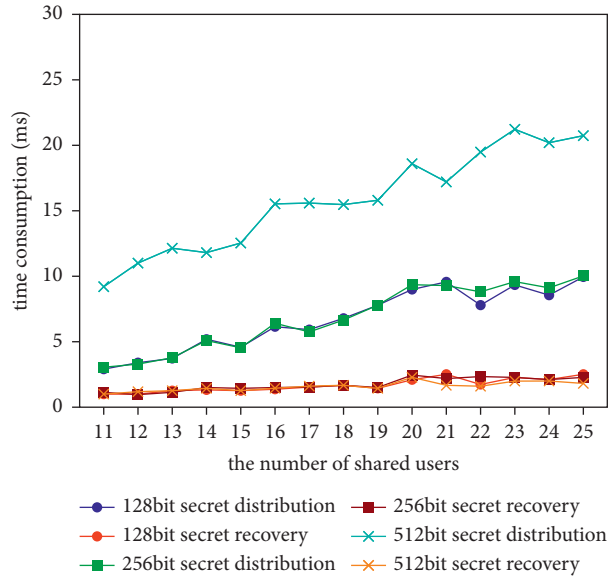


FIGURE 5: Secret distribution and recovery time.

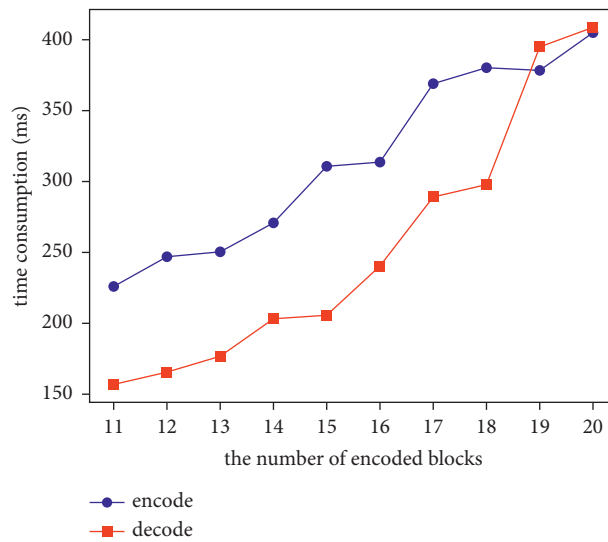


FIGURE 6: Erasure code encoding and decoding time.

literature [29] is a multi-share, single-user mode. That is, the key shares are transferred to n nodes or objects. The user applies for and obtains more than t valid key shares to recover the data, and our scheme adopts a multi-share, multi-user mode, which directly distributes the key shares to the data inquirers who have passed the review of the data applicant. As long as the number of data inquirers applied for viewing exceeds t or the data owner agrees, the key can be reconstructed. Therefore, the single transmission volume of our scheme is $O(t)$ level, which is higher than that $O(1)$ level

of the multi-share, single-user query mode. But the total transmission volume of our scheme is only $n \times (t + 1)$, lower than the total transmission volume $n \times 2t$ of the multi-share single-user query mode. In addition, our scheme uses the coefficient matrix of the Blakley space plane equations to segment the ciphertext data and reduce the data size to facilitate the chaining. At the same time, it increases the power of data reconstruction during query. However, the solution using Lagrange interpolation can only achieve data segmentation through additional mechanisms.

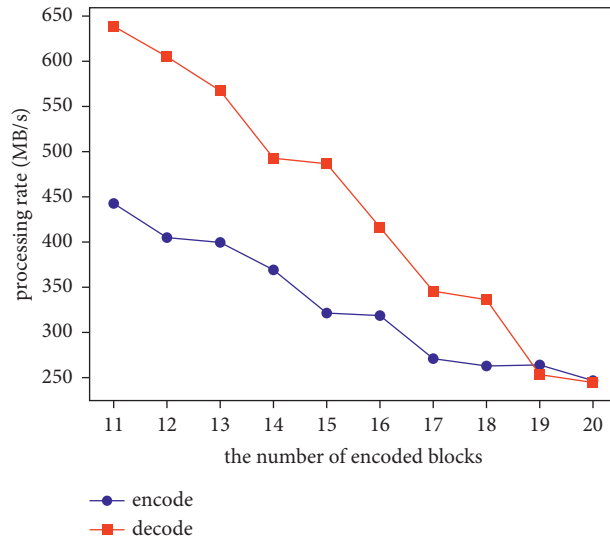


FIGURE 7: Erasure code encoding and decoding rate.

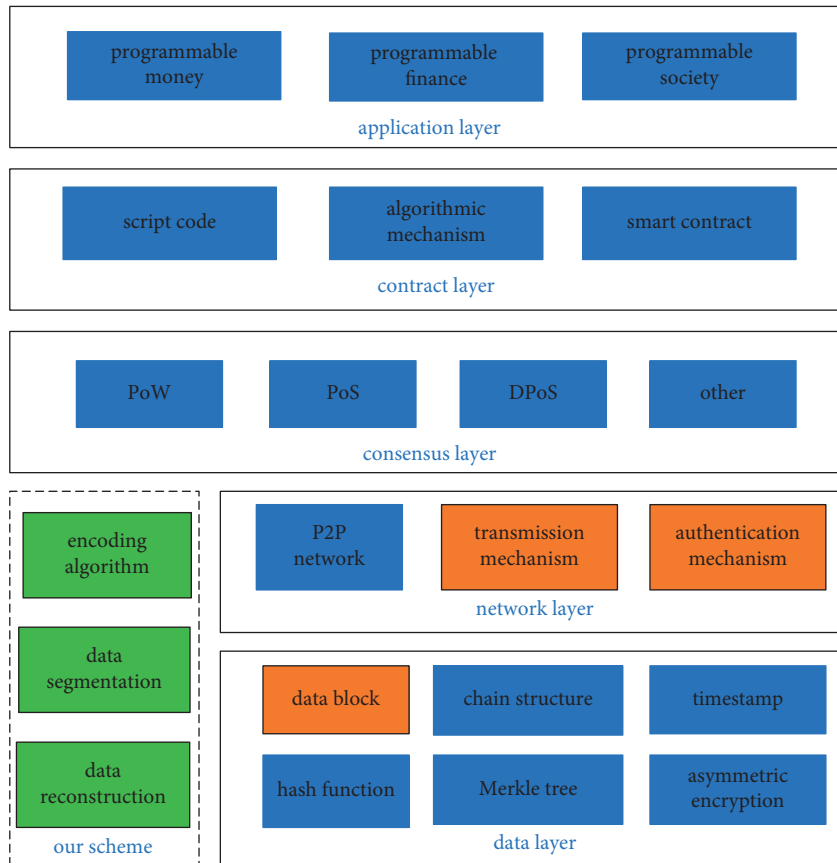


FIGURE 8: The relationship between this scheme and the blockchain architecture.

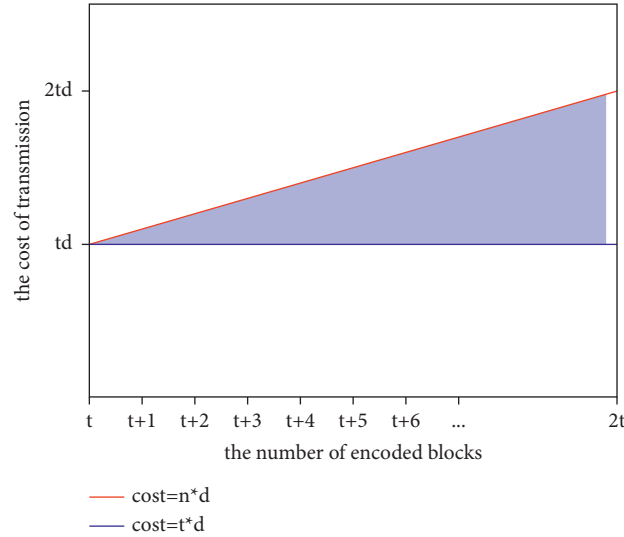


FIGURE 9: Network layer data transmission costs.

TABLE 2: Comparison of schemes.

	Our scheme	Literature [28]	Literature [29]
Query mode	Multi-share, multi-user	Multi-share, single-user	Multi-share, single-user
Single transmission volume	$O(t)$	$O(1)$	$O(1)$
Total transmission volume	$n \times (t + 1)$	$n \times 2t$	$n \times 2t$
Data reconstruction	RS code	Not supported	Not supported
Depends on a specific platform	No	No	Ethereum
Applicable scene	Data recording and management	Cyber-physical system	Cloud data protection

6. Conclusion

In view of the blockchain data query requirements in the data recording and management scenario, this study analyzes the phenomenon that the data on the chain are too large to be directly stored in the block and proposes a blockchain data sharing query scheme based on threshold secret sharing. The sharing query scheme uses the Blakley space plane equation to share the secret elements used for data query. It restricts the rights of blockchain data inquirers, thus improving the security of blockchain data queries. At the same time, a method for uploading data to the blockchain based on erasure codes is proposed. It uses the Blakley space plane equation coefficient matrix as the erasure code encoding matrix to segment the ciphertext data. It not only reduces the data size but also improves the system error tolerance rate during query. The simulation experiment results show that the additional time consumption of the secret-sharing recovery and data segmentation reconstruction of this solution is much less than the cost of block generation. Therefore, it will not affect the normal operation of the blockchain application in the scene of data recording and management, and it can also improve the security of blockchain application data.

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 Nature Science Foundation of China (nos. 61972208 and 61672299), Post-graduate Research & Practice Innovation Program of Jiangsu Province(SJKY19_0770).

References

- [1] S. Nakamoto, "A. Bitcoin: A peer-to-peer electronic cash system," 2009, <http://bitcoin.org/bitcoin.pdf>.
- [2] Z. Yu, X. G. Liu, and G. Wang, "A survey of consensus and incentive mechanism in blockchain derived from P2P," in *Proceedings of the . 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 1010–1015, Singapore, December 2018.
- [3] L. Chen, F. Xiang, and Z. X. Sun, "A survey of blockchain security technologies based on attribute-based cryptography," *Acta Electronica Sinica*, vol. 49, no. 1, pp. 192–200, 2021.
- [4] G. Ma, C. Ge, and L. Zhou, "Achieving reliable timestamp in the bitcoin platform," *Peer-to-Peer Networking and Applications*, vol. 13, no. 6, pp. 2251–2259, 2020.
- [5] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, and D. Niyato, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, Article ID 22370, 2019.

- [6] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," 2015, <http://gavwood.com/Paper.pdf>.
- [7] E. Androulaki, A. Barger, V. Bortnikov et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the The Thirteenth EuroSys Conference*, pp. 1–15, Porto, Portugal, April 2018.
- [8] Y. Yuan and F. Y. Wang, "Blockchain: the state of the art and future trends," *Acta Automatica Sinica*, vol. 42, no. 4, pp. 481–494, 2016.
- [9] S. Q. Zeng, R. Huo, T. Huang, J. Liu, S. Wang, and W. Feng, "Survey of blockchain: principle, progress and application," *Journal on Communications*, vol. 41, no. 1, pp. 134–151, 2020.
- [10] G. Yu, T. Z. Nie, X. H. Li, Y. F. Zhang, D. R. Shen, and Y. B. Bao, "The challenge and prospect of distributed data management techniques in blockchain systems," *Chinese Journal of Computers*, vol. 42, pp. 1–27, 2019.
- [11] A. D. Liu, X. H. Du, N. Wang, and S. Z. Li, "Research progress of blockchain technology and its application in information security," *Ruan Jian Xue Bao/Journal of Software*, vol. 29, no. 7, pp. 2092–2115, 2018.
- [12] H. Li, D. Han, and M. Tang, "Logisticschain: a blockchain-based secure storage scheme for logistics data," *Mobile Information Systems*, vol. 2021, Article ID 8840399, 15 pages, 2021.
- [13] J. Tian, X. Jing, and R. Guo, "Public audit scheme of shared data based on blockchain," *Communications in Computer and Information Science*, vol. 1105, pp. 327–344, 2019.
- [14] L. Wang, W. Liu, and X. Han, "Blockchain-based government information resource sharing," in *Proceedings of the 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 804–809, Shenzhen, China, December 2017.
- [15] X. Zhang and Y. Yin, "Research on digital copyright management system based on blockchain technology," in *Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 2093–2097, Chengdu, China, June 2019.
- [16] L. Ouyang, S. Wang, Y. Yuan, X. Ni, F. Y. Wang, and X. Han, "Smart contracts: archit and research progresses," *Acta Automatica Sinica*, vol. 45, no. 3, pp. 445–457, 2019.
- [17] C. Zhang, Q. Li, Z. H. Chen, Z. R. Li, and Z. Zhang, "Medical chain: alliance medical blockchain system," *Acta Automatica Sinica*, vol. 45, no. 8, pp. 1495–1510, 2019.
- [18] S. Wu and J. Du, "Electronic medical record security sharing model based on blockchain," in *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, pp. 13–17, Kuala Lumpur, Malaysia, January 2019.
- [19] H. B. Tan, T. Zhou, H. Zhao et al., "Archival data protection and sharing method based on blockchain," *Ruan Jian Xue Bao/Journal of Software*, vol. 30, no. 9, pp. 2620–2635, 2019.
- [20] V. Jaiman and V. Urovi, "A consent model for blockchain-based health data sharing platforms," *IEEE Access*, vol. 8, pp. 143734–143745, 2020.
- [21] J. R. Wang, S. Z. Yu, and R. Li, "Medical blockchain of privacy data sharing model based on ring signature," *Journal of University of Electronic Science and Technology of China*, vol. 48, no. 6, pp. 886–892, 2019.
- [22] A. K. Shrestha and J. Vassileva, "User data sharing frameworks: a blockchain-based incentive solution," in *Proceedings of the 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0360–0366, Vancouver, Canada, October 2019.
- [23] S. Huang, L. W. Chen, and B. B. Fan, "Data security sharing method based on CP-ABE and blockchain," *Computer Systems & Applications*, vol. 28, no. 11, pp. 79–86, 2019.
- [24] X. L. Wang, X. Z. Jiang, and Y. Li, "Model for data access control and sharing based on blockchain," *Ruan Jian Xue Bao/Journal of Software*, vol. 30, no. 6, pp. 1661–1669, 2019.
- [25] T. T. Thwin and S. Vasupongayya, "Blockchain based secret-data sharing model for personal health record system," in *Proceedings of the 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, pp. 196–201, Krabi, Thailand, August 2018.
- [26] H. T. T. Truong, M. Almeida, G. Karame, and C. Soriente, "Towards secure and decentralized sharing of IoT data," in *Proceedings of the .2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 176–183, Atlanta, USA, July 2019.
- [27] M. J. Gao and H. Q. Wang, "Blockchain-based searchable medical data sharing scheme," *Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition)*, vol. 39, no. 6, pp. 94–103, 2019.
- [28] Z. Xu, J. Yang, and J. Yin, "A lightweight data sharing mechanism and multiparty computation for CPS," in *Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–5, Antwerp, Belgium, May 2020.
- [29] N. Sohrabi, X. Yi, Z. Tari, and I. Khalil, "BACC: blockchain-based access control for cloud data," in *Proceedings of the Australasian Computer Science Week Multiconference*, pp. 1–10, Melbourne VIC Australia, February, 2020.
- [30] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [31] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the 1973 Managing Requirements Knowledge, International Workshop on. IEEE Computer Society*, p. 313, New York, NY, USA, June 1979.
- [32] S. Johnson, "Burst erasure correcting LDPC codes," *IEEE Transactions on Communications*, vol. 57, no. 3, pp. 641–652, 2009.
- [33] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.
- [34] R. M. Roth and A. Lempel, "On MDS codes via Cauchy matrices," *IEEE Transactions on Information Theory*, vol. 35, no. 6, pp. 1314–1319, 1989.

Research Article

A Blockchain-Based User Authentication Scheme with Access Control for Telehealth Systems

Shuyun Shi,^{1,2} Min Luo ,^{1,3} Yihong Wen,⁴ Lianhai Wang,³ and Debiao He ^{1,5}

¹School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

²Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

³Shandong Provincial Key Laboratory of Computer Networks, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China

⁴54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China

⁵Shanghai Key Laboratory of Privacy-Preserving Computation, MatrixElements Technologies, Shanghai 201204, China

Correspondence should be addressed to Min Luo; mluo@whu.edu.cn

Received 20 October 2021; Accepted 7 February 2022; Published 30 March 2022

Academic Editor: Mamoun Alazab

Copyright © 2022 Shuyun 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.

With the development of telecommunication systems and customized monitoring devices, telehealth has been widely used to improve medical quality and reduce overall health costs. However, the convenience of connection between the providers and patients through a public channel also leads to significant security and privacy concerns. Though there have been many authentication schemes designed for secure communications in telecare systems, most of them suffer from malicious attacks or have heavy computation and communication costs. Thus, in this article, we proposed a blockchain-based user authentication scheme integrating with access control and physical unclonable function (PUF). Permissioned blockchain and PUF are used to support secure data sharing across the healthcare service providers and identify the devices, respectively. Security analysis shows that our protocol satisfies the security requirements for telehealth services and is provably secure in the random oracle model. The performance evaluation demonstrates that it has less computation and communication costs compared with three of the latest schemes.

1. Introduction

Telecare medical information systems (TMIS) support remote medical services by providing online patient diagnosis to reduce healthcare service costs and improve patient health outcomes. The COVID-19 pandemic has promoted the use of telehealth to deliver, which can interrupt the transmission of the disease and facilitate public health mitigation by reducing outings.

As shown in Figure 1, patients at home are equipped with wearable monitoring devices. These devices can continually collect and transmit health data (e.g., blood pressure, blood sugar, heart rate, and more) to mobile devices. Health data will be transmitted to healthcare service providers (e.g., hospitals and health authorities) over the open internet. Then, care staff (e.g., doctors and nurses) can monitor the

patient's condition remotely and make timely treatment decisions for better outcomes.

Vulnerabilities in the wireless networks offer some easy entry points for malicious adversaries, yet these networks are important for connections between patients at home and remote healthcare organizations. Many countries have laws that are designed to protect the patient's privacy, such as the Health Insurance Portability and Accounting Act (HIPAA) in the United States. Security in telecare services, i.e., how to ensure patient data security and privacy during transmission through the public channel, becomes a significant concern [1, 2].

User authentication is a necessary first step to ensure that only authorized users have access to protected data. Password-based user authentication scheme as the most convenient mechanism is widely employed, however, it is vulnerable to various attacks and could be a threat to data

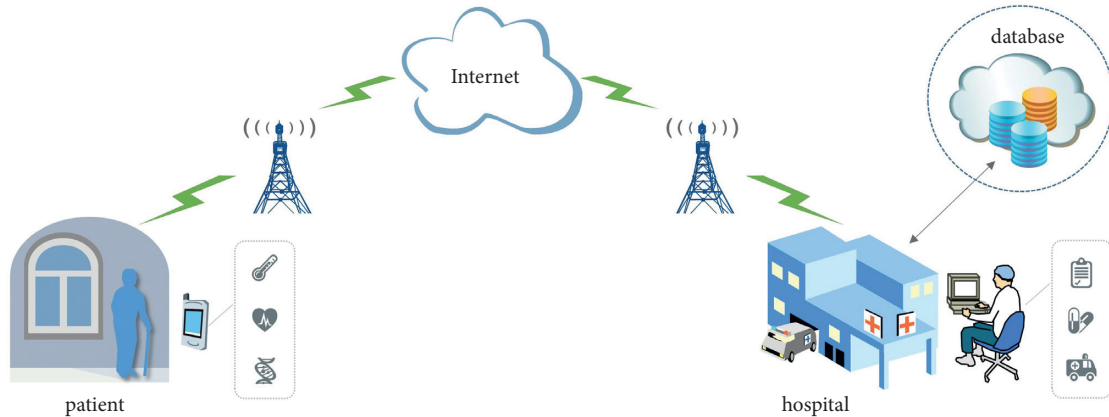


FIGURE 1: General architecture for telehealth system.

security. Multifactor authentication is a recommended approach in which any user is granted access to certain data after validating two or more pieces of evidence the user has.

In the meantime, people's mobility between different healthcare institutions raises some problems: user authentication in multiple servers and secure data sharing across different servers. To address the above issues, multiserver authentication is used for access to multiple servers with one credential [3] in existing schemes. However, most schemes [4–6] have unsatisfactory performance or may suffer serious security problems in the telehealth services environment.

Furthermore, interaction with patient records from personal devices may be fraught with peril since it is difficult for healthcare servers to verify whether those devices meet security configuration requirements. Attackers could impersonate legal users for free treatment or profit once they guess the correct password. Otherwise, they could impersonate the healthcare providers to offer false treatment, which will cause a critical medical accident.

Therefore, it is necessary to design a new authentication scheme to ensure security and data sharing and resist various attacks in the telecare services environment.

1.1. Our Contribution. In this paper, we propose a multi-server authentication scheme with the integration of user authentication and access control, which determines the access control to achieve selective data disclosure, enhance data privacy, and improve the scheme's efficiency.

Blockchain has been widely applied in different areas because of its properties of immutability and decentralization [7–9]. In our scheme, some registration information will be recorded in the blockchain served as a trusted bulletin board to achieve secure data sharing across different healthcare provider servers.

PUF has been a promising cryptographic primitive, and it has been demonstrated in the security domain as well [10]. It serves as one of the authentication factors to identify the devices in the IoT systems, with the characteristics of being unclonable, unpredictable, and computable. We will introduce this technology to resist various attacks (e.g., impersonation attacks and physical attacks) for telehealth services.

The major contributions of this paper are summarized as follows:

- (i) Firstly, by leveraging blockchain technology and PUFs, we design an efficient user authentication protocol with access control for the telehealth system.
- (ii) Secondly, a comprehensive security analysis is given to show that the proposed protocol is provably secure and can satisfy the security requirements in the telehealth system.
- (iii) Finally, we implement a prototype by smart contract based on Hyperledger Fabric. We analyze the performance to show that our scheme has less computation and communication costs than previously proposed protocols.

1.2. Organization of Our Paper. The paper is organized as follows: in section 2, we discuss some related literature. In section 3, we introduce the preliminaries used in this paper. In sections 4 and 5, we present the details of the system framework and proposed authentication protocol for telecare service. In sections 6 and 7, we give provable security in the random oracle model and evaluate the performance of the proposed scheme. Finally, section 8 concludes this paper.

2. Related Work

To secure remote healthcare services, some authentication schemes have been proposed. Debiao et al. [11] proposed an improved scheme to overcome the weakness of Wu et al.'s scheme [12] for TMIS. Their scheme is more efficient and applies to low-power mobile devices. However, the scheme cannot resist the offline password attack. Kumari et al. [13] proposed an improved user authentication scheme for applications in TMIS. Chen et al. [14] proposed a medical data exchange protocol based on the cloud environment for medical advice. Patient inspection information can be protected by asymmetric encryption. However, the scheme cannot provide user anonymity and has heavy computation costs.

Chiou et al. [15] resolved these security problems and provided a complete system implementation. Mohit et al. [4] reviewed Chiou et al.'s [15] protocol and found that it is susceptible to user anonymity and some attacks. They designed a lightweight authentication scheme in the cloud environment for TMIS. However, Kumar et al. [5] found that Mohit et al.'s [4] scheme is vulnerable to various attacks and cannot provide user anonymity and session key protection. They proposed an improved protocol for single-server architecture in TMIS but could not satisfy the perfect forward secrecy or multiserver environment.

Multiserver authentication scheme was first proposed by [3] using a neural network. Because of the complexity of the neural network, lots of schemes [16–19] were proposed to improve the performance and enhance the security. Multiserver authentication scheme without online RC [20, 21] is suitable for various applications, which reduces the cost of trusted RC establishment and authentication communication.

Recently, blockchain has become a research hotspot in telemedicine to ensure healthcare data security and to support data sharing. Liu et al. [22] proposed privacy-preserving mutual authentication in TMIS, which provides user anonymity and malicious users traceability if necessary. Yazdinejad et al. [23] designed a blockchain-based authentication scheme to reduce reauthentication across different hospitals, which can increase throughput and reduce time overhead for resource-limited devices. Li et al. [24] proposed a group authentication mechanism for authorized group members to access sensitive health records in the remote medical monitoring scene. Cheng et al. [25] designed a multiple identity authentication for a secure medical data sharing model based on blockchain to avoid over-reliance on a third party. Lin et al. [26] designed a transitively closed undirected graph authentication scheme for dynamic blockchain-based identity management systems, which is efficient for signers to update their certificates without signing again. Wang et al. [27] proposed a decentralized, secure, and lightweight certificateless signature (CLS) protocol by transforming the logic of KGC into smart contract code, which can resist key generation center compromised attacks and distributed denial of service (DDoS) attacks. Xiong et al. [28] presented ECDSA batch verification protocol in the blockchain-enabled Internet of Medical Things (IoMT) to enhance authentication efficiency and support identification algorithms for false signatures.

Nevertheless, all of the above schemes have no consideration for integration authentication with access control to improve the system efficiency. The idea of the integration of user authentication and access control was first proposed by Harn and Lin [29] to avoid potential security problems between these two protection mechanisms. Later, some improved protocols [30–32] were designed to enhance security and implement a protection scheme in distributed systems. Recently, Lin et al. [33] presented a remote mutual authentication scheme with fine-grained access control

based on blockchain for industry 4.0 deployment. Xiong et al. [34] proposed an integrated scheme for a mobile cloud environment (MCC) without the trusted party to store the access control list. However, the computational overhead is expensive for limited mobile devices in telemedicine services.

Additionally, many PUF-based authentication schemes have been proposed for IoT systems, wireless sensor networks (WSNs), and so on. Since smart cards/devices are not tamper-resistant, some two-factor authentication schemes are susceptible to various physical attacks. To address the issues, PUFs are used as one of the authentication factors presented in some literature [35–38] to enhance the properties of lightweight authentication solutions.

In this paper, we will construct a blockchain-based user authentication scheme for better efficiency and security, in which access control integration can enhance the authentication efficiency, and a physical unclonable function is applied to identify the devices against various attacks in the telehealth environment.

3. Preliminaries

3.1. Bilinear Pairings. Let p, q be large prime numbers. Let G_1 be a cyclic additive group and G_2 be a multiplicative group with the same order q . $e: G_1 \times G_1 \rightarrow G_2$ denotes a bilinear map, where a generator P of G_1 and a generator $g = e(P, P)$ of G_2 , when the subsequent conditions meet.

- (i) *Bilinear:* $e(aP, bQ) = e(P, Q)^{ab}$ for all $a, b \in Z_q^*$ and $P, Q \in G_1$.
- (ii) *Nondegeneracy:* there exists an element $P \in G_1$ such that $e(P, P) \neq 1_{G_2}$.
- (iii) *Computability:* given any two elements $P, Q \in G_1$, it is efficient to compute $e(P, Q)$.

The following mathematical problems are difficult, i.e., there is no polynomial algorithm to solve will be used in our proposed scheme.

- (i) *Discrete Logarithm (DL) Problem:* given an element $Z \in G_1$ ($z \in G_2$), find α such that $Z = \alpha P$ ($z = g^\alpha$).
- (ii) *Computational Diffie-Hellman (CDH) Problem:* given two elements $xP, yP \in G_1$ ($g^x, g^y \in G_2$), where $x, y \in Z_q^*$ are unknown, calculate $xyP \in G_1$ ($g^{xy} \in G_2$).
- (iii) *Modified Bilinear Inverse Diffie-Hellman with k Value (k-mBIDH) Problem:* given k elements $a_1, a_2, \dots, a_k \in Z_q^*$ and $k+2$ elements $\widehat{s}P, \widehat{t}P, (1/\widehat{s} + a_1)P, (1/\widehat{s} + a_2)P, \dots, (1/\widehat{s} + a_k)P \in G_1$, where $\widehat{s}, \widehat{t} \in Z_q^*$ is unknown, calculate $e(P, P)^{\widehat{t}/\widehat{s}+a}$ for any $a \notin \{a_1, a_2, \dots, a_k\}$.

3.2. Chinese Remainder Theorem (CRT). Given N coprime integers m_1, m_2, \dots, m_k , where $\gcd(m_i, m_j) = 1$ for $i, j = 1, 2, \dots, k$ and $i \neq j$. For integer a_i , there exists the following:

$$\begin{cases} X \equiv a_1 \pmod{m_1} \\ X \equiv a_2 \pmod{m_2} \\ \vdots \\ X \equiv a_k \pmod{m_k} \end{cases} \quad (1)$$

The common solution X can be computed as follows:

$$X \equiv \sum_{i=1}^k \frac{M}{m_i} \cdot e_i \cdot a_i, \quad (2)$$

where $M = \prod_{i=1}^k m_i$ and $M/m_i \cdot e_i \equiv 1 \pmod{m_i}$ for $i = 1, 2, \dots, k$.

3.3. Physical Unclonable Function. The physical unclonable function is a one-way mapping from a challenge space \mathbb{C} to a response space \mathbb{R} based on the unclonable characteristic of the underlying physical device. A set of challenge-response pairs (CRPs) is unique for each device, which can be used to identify the device. A PUF circuit must meet the properties below:

- (i) *Unpredictability*: given any challenge \mathcal{C} , the probability to evaluate the response \mathcal{R} of PUF is negligibly small without PUF instance.
- (ii) *Computability*: given any challenge \mathcal{C} , it is easy to evaluate the response \mathcal{R} for any PUF instance.
- (iii) *Uniqueness*: for any two PUF_1 and PUF_2 , given the same challenge \mathcal{C} , the probability to evaluate the same response $PUF_1(\mathcal{C}) = PUF_2(\mathcal{C})$ is negligibly small.
- (iv) *One-way*: given any $\mathcal{R} \in \mathbb{R}$, there exists no polynomial algorithm $\text{InversePUF}: \mathcal{R} \rightarrow \mathcal{C}$ for any challenge $\mathcal{C} \in \mathbb{C}$.

3.4. Fuzzy Extractor. The PUF circuit is susceptible to interference to generate the noisy response with low entropy, which may fail to authenticate the device. Fuzzy extractor has been developed to recover a reliable high-entropy response from a noisy response to enhance the security of authentication. Fuzzy extractor consists of the following two algorithms:

- (i) *Gen*(\cdot): a probabilistic key generation algorithm $(\kappa, h, d) = \text{Gen}(R)$ to generate the key κ and helper data h, d .
- (ii) *Rec*(\cdot): a deterministic reconstruction algorithm $(\kappa) = \text{Rec}(R', h, d)$ to recover the key κ from a noisy response R' and helper data h, d , which the Hamming distance between the original response R and the noisy response R' is at most d .

3.5. Blockchain and Smart Contract. Blockchain technology is an immutable and distributed data ledger consisting of an append-only sequence of blocks chained by the cryptographic hash function. Based on permission authorized to network nodes, blockchain platforms are divided into three

types: private blockchain, consortium blockchain, and public blockchain. In the proposed scheme, Hyperledger Fabric is chosen to support the flexible transaction and Turing-complete smart contracts.

The smart contract is an autoexecuted program deployed in the blockchain network, which can achieve complex functions, and it can be invoked by authorized nodes sending a legal transaction. The transaction is contained in the block after verification by the nodes. In our proposed scheme, we design the smart contract for registration information sharing between a trusted register center and multiple healthcare servers. The register center and servers can upload, query, and update the information by sending a transaction signed by its private key to invoke the smart contract. Users can query the information to ensure that their information can be authenticated by the whole network.

4. System Framework and Security Model

4.1. Network Model. Figure 2 describes the network model that consists of four types of entities in our proposed scheme, namely the trusted registration center (RC), the permissioned blockchain network (BC), the servers (hospital and medical institutions), and the mobile users (patients and healthcare servicers).

- (i) *RC*: it is a trusted third party and is responsible for system initialization, user/server registration, and registration information recorded in the blockchain network. It generates a master private key and issues a private key of the user/server according to their identities. Besides, it also randomly generates a challenge and sends it to the user's devices. The registration record and challenge-response pair will be uploaded in the permissioned blockchain.
- (ii) *BC*: it acts as a trusted recorder for registration, sharing, and updating by the smart contract. The trusted RC and servers join as network nodes to maintain the permissioned blockchain network together.
- (iii) *Healthcare Servers*: the servers provide data storage for patient health records from wearable devices and support remote healthcare services between the healthcare provider and patients at home.
- (iv) *Mobile Users*: there are two types of mobile users: patients and healthcare providers. We assume that all the mobile devices are equipped with a PUF and fuzzy extractor. The output of the PUF is used as one of the authentication factors. Moreover, a fuzzy extractor has been employed to recover the noisy PUF. The mobile devices of patients can upload biomedical data collected by wearable devices to remote servers for personalized medical services after mutual authentication. The mobile devices of healthcare providers can get patient conditions and offer clinical diagnostics after authentication.

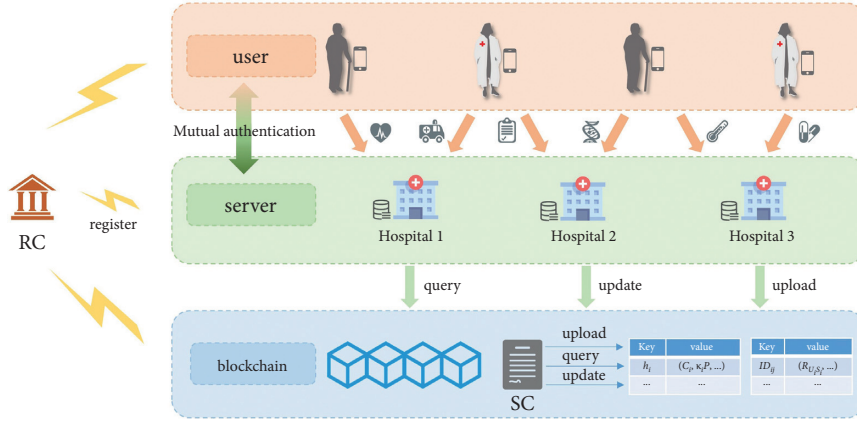


FIGURE 2: System model for remote healthcare service.

4.2. *Network Assumptions.* Patients can enjoy remote medical service by off-chain subscription service that hospitals/medical institutions provide and generate the access control list representing patients' service time. Healthcare providers should also get permission (e.g., read, write, delete) to operate the data to achieve diverse data sharing across different institutions.

Each subscription service maps to a specific value according to the mapping rules AC_{S_i} , predefined and will be stored in the blockchain by each server, which can be retrieved by any server to check the validity. The access control list is represented by $ACL = \langle S, R \rangle$, where $S = \{S_1, S_2, \dots, S_n\}$, $R = \{R_{S_1}, R_{S_2}, \dots, R_{S_n}\}$, where R_{U_i, S_j} represents the access permission the user applies for in S_i , as shown in Tables 1 and 2. The server predefines the mapping rules AC_{S_j} of access permission, as shown in Table 3, and sends it to the RC in the registration phase.

Following the research efforts [39], after mapping, RC calculates the common solution M of each user using CRT as follows:

$$\begin{cases} M \equiv \text{Num}_{R_1} \pmod{N_{S_1}} \\ M \equiv \text{Num}_{R_2} \pmod{N_{S_2}} \\ \vdots \\ M \equiv \text{Num}_{R_k} \pmod{N_{S_n}} \end{cases} \quad (3)$$

In the authentication phase, the specific access permission can be calculated by the common solution.

$$\text{Num}_{R_i} \equiv M \pmod{N_{S_i}}, \quad (4)$$

then the server can check whether the user has been authorized or expired by equation (4).

4.3. *Security Requirements.* To ensure the security of the remote healthcare service, the proposed scheme should satisfy the following requirements [21, 34–36, 40]:

Single registration: for convenience to users, the proposed scheme should provide single registration. Users only register with the trusted RC once before they can communicate with healthcare servers.

TABLE 1: The access control list of patients.

S_i	S_1	S_2	S_3	...	S_n
R_{U_i, S_j}	One month	One month	Three months	...	Seven months

TABLE 2: The access control list of healthcare providers.

S_i	S_1	S_2	S_3	...	S_n
R_{U_i, S_j}	Read	Read write	Delete	...	Read Write Delete

No online registration center: the proposed scheme should achieve mutual authentication without RC to relieve the communication overhead and resist the single point of failure.

Mutual authentication: to ensure that legal users and servers could access healthcare data, the proposed scheme should provide mutual authentication between U_i and S_j to verify the legality of each other.

User anonymity: to preserve the privacy of users, the proposed scheme should protect user identity. Any adversary cannot extract the real identity from the intercepted message.

Untraceability: for the better protection of user privacy, no potential adversary can trace the user's activities and behavior patterns by analyzing the intercepted message.

Session key agreement: the proposed scheme should generate the session key between the users and servers to encrypt the message in future communications.

Perfect forward secrecy: the proposed scheme should provide perfect forward secrecy to ensure the security of messages in the previous sessions. No potential attackers can generate the session key of previous sessions even if they obtain the long-term private key of two participants.

Two-factor security: the proposed scheme should provide two-factor security, i.e., password and mobile device embedded with PUF.

TABLE 3: The mapping rules of access permission.

Permission	One month	Three months	Five months	Seven months	Only read	Read write	Read write delete
Numbers	3	4	5	7	11	13	17

Access control for data privacy: the healthcare service providers can provide personalized remote healthcare services that users subscribe to for a specific service time. In the meantime, healthcare providers can be authorized to access the data with different permissions for preserving data privacy.

Resistance of various attacks: to resist known attacks existing in the service system, the proposed scheme should resist known attacks, including impersonation attacks, physical attacks, replay attacks, man-in-the-middle attacks, etc.

5. The Proposed Scheme

We describe the proposed scheme in this section. The proposed scheme consists of seven phases: blockchain initialization, system setup phase, server registration phase, user registration phase, mutual authentication phase, password update phase, and access rights update phase.

5.1. Blockchain Initialization. RC establishes a consortium blockchain (e.g., Hyperledger Fabric) among RC and servers as network nodes to maintain the blockchain. The servers must register and enroll the identities as legitimate members to engage in the consensus process.

In the meantime, the smart contract (SC) will be deployed in the blockchain network and can be invoked by transaction. Algorithms 1 and 2 show that the smart contract supports the upload and query of challenge-response pair (CRP) and access permission. In the subscription phase, the servers will upload the subscription service time (for patients) or service permission (for healthcare providers) into the blockchain, which can be validated when RC computes users' common solution. In the registration phase, RC will randomly generate the challenge and then receive the response produced by user's fuzzy extractor via a secure channel. The challenge-response pair will be stored in the ledger to share with servers in the authentication process.

5.2. System Setup Phase. In this phase, the trusted RC generates system parameters and selects the master private key.

- (1) RC selects additive group G_1 and multiplicative group G_2 with the same prime order q and a bilinear pairing $e: G_1 \times G_1 \rightarrow G_2$. RC also chooses a generator P of G_1 and then computes $g = e(P, P)$ of G_2 .
- (2) RC randomly chooses $s \in Z_q^*$ as the master private key and calculates its public key $P_{\text{pub}} = sP$.
- (3) RC selects seven secure hash functions $h_0: \{0, 1\}^* \rightarrow \{0, 1\}^{l_0}$, $h_1: \{0, 1\}^* \times G_1 \rightarrow Z_q^*$,

$$h'_1: \{0, 1\}^* \rightarrow Z_q^*, h_2: \{0, 1\}^* \times \{0, 1\}^* \times G_1 \times \{0, 1\}^* \times G_1 \times \{0, 1\}^* \rightarrow \{0, 1\}^*, h_3: G_2 \rightarrow \{0, 1\}^*, h_4: \{0, 1\}^* \times G_1 \times \{0, 1\}^* \rightarrow Z_q^*, h_5: \{0, 1\}^* \times \{0, 1\}^* \times \{0, 1\}^* \times \{0, 1\}^* \times G_1 \times G_1 \times G_1 \times \{0, 1\}^* \rightarrow \{0, 1\}^*, h_6: G_1 \times G_1 \times G_1 \times G_1 \times \{0, 1\}^* \times \{0, 1\}^* \times \{0, 1\}^* \rightarrow Z_q^*$$

- (4) RC publishes system parameters $\text{params} = \{G_1, G_2, P, e, q, g, P_{\text{pub}}, h_0, h_1, h'_1, h_2, h_3, h_4, h_5, h_6\}$ and keeps $\{s\}$ securely.

5.3. Server Registration Phase. As shown in Figure 3, the server S_j registers with the RC to obtain its long-term private key through a secure channel. The steps below will be executed between S_j and RC.

- (1) The server S_j selects its ID_{S_j} and predefines the access control mapping rules AC_{S_j} . Then, S_j transmits them to RC.
- (2) RC computes $\mathcal{D}_{S_j} = 1/s + h'_1(ID_{S_j}) \cdot P$, selects a coprime integer N_{S_j} , and sends $\{\mathcal{D}_{S_j}, N_{S_j}\}$ to S_j . RC stores $\{S_j, N_{S_j}, AC_{S_j}\}$ securely.
- (3) After receiving, S_j keeps $\{\mathcal{D}_{S_j}, N_{S_j}\}$ secretly.

5.4. User Registration Phase. There are two types of mobile users in this scheme, including patients (remote service subscribers) and healthcare providers (remote service providers). The following steps will be executed by mobile users to register with RC, as shown in Figure 4.

- (1) The mobile user U_i selects an identity ID_i and a password PW_i and generates a nonce n_i . After subscription service, the access control list $ACL_i = \langle S_j, R_{U_i} \rangle$ and $P_i = h_0(ID_i \| PW_i \| n_i)$ will be sent to RC through a secure channel.
- (2) After receiving the request, RC computes the common solution M using CRT according to the access control list of user $R_{U_i} = \{R_{U_i, S_1}, R_{U_i, S_2}, \dots, R_{U_i, S_n}\}$, mapping rules $\{AC_{S_1}, AC_{S_2}, \dots, AC_{S_n}\}$, and $N = \{N_{S_1}, N_{S_2}, \dots, N_{S_n}\}$ by equation (3). After that, RC generates a random number $r_i \in Z_q^*$, $R_i = r_i P$, and calculates $\mathcal{D}_i = (r_i + sh_i) \bmod q$ using $h_i = h_1(ID_i \| M \| R_i)$ and its master private key. RC also generates a random challenge C_i and sends it to U_i .
- (3) After receiving the challenge C_i , U_i extracts the outputs of the PUF $\gamma_i = PUF(C_i)$ and then obtains the secret key κ_i and helper data hd from $FE.Gen(\gamma_i)$ using a fuzzy extractor. After that, U_i sends $\{\kappa_i, hd\}$ to RC for authentication in future.

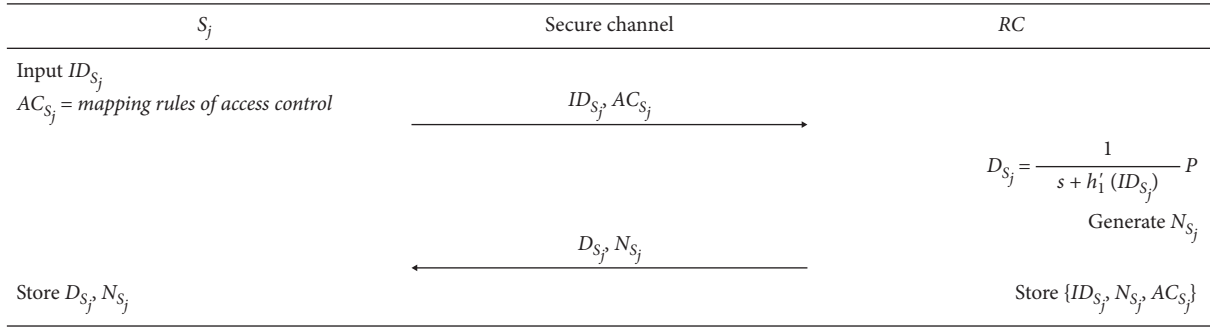


FIGURE 3: Server registration.

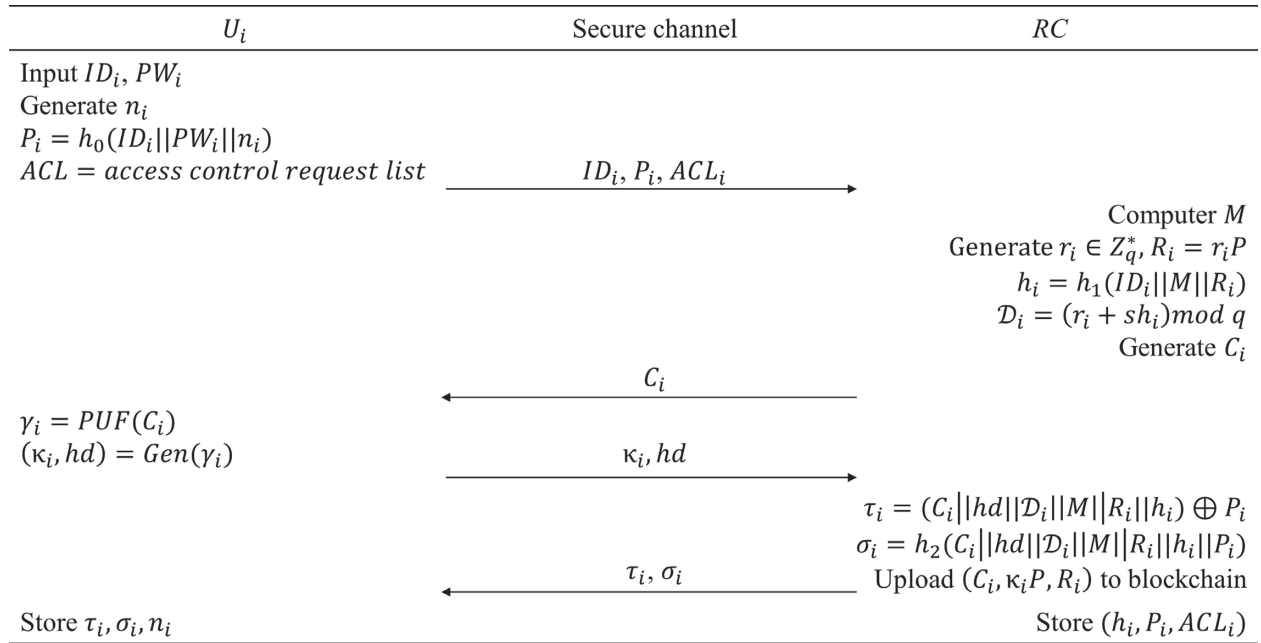


FIGURE 4: Mobile user registration.

- (4) RC computes $\tau_i = (C_i || hd || \mathcal{D}_i || M || R_i || h_i) \oplus P_i$, $\sigma_i = h_2(C_i || hd || \mathcal{D}_i || M || R_i || h_i || P_i)$, and $\kappa_i P$ and then uploads the “challenge-response” pair $\{C_i, \kappa_i P, R_i\}$ in the blockchain to share with the servers (healthcare providers), where t_s denotes the service start time as shown in Algorithm 2. After that, RC sends $\{\tau_i, \sigma_i\}$ to U_i and stores the user registration table $\{ID_i, P_i, ACL_i\}$ in the blockchain.
- (5) U_i stores $\{\tau_i, \sigma_i, n_i\}$ in the secure memory.

5.5. Authentication Phase. As depicted in Figure 5, the user U_i authenticates with the server S_j , and then a common session key SK is generated for secure communications.

- (1) U_i inputs his identity ID_i and password PW_i . His/Her mobile device computes $P_i = h_0(ID_i || PW_i || n_i)$, $C_i = h(d || \mathcal{D}_i || M || R_i || h_i || P_i)$, $\tau_i = \tau_i \oplus P_i$, $\sigma_i = h_2(C_i || hd || \mathcal{D}_i || M || R_i || h_i || P_i)$ and then checks whether the equation $\sigma_i' = \sigma_i$ holds. If not, it terminates the request. Otherwise, the device extracts the PUF output $\gamma_i' = PUF(C_i)$ and $\kappa_i = \text{Rec}(\gamma_i', hd)$ using a fuzzy

extractor. After that, it chooses a random number $x \in Z_q^*$ and calculates $X' = g^x$, $X = xP$, $U_1 = x(P_{\text{pub}} + h'_1(ID_{S_j})P)$, $N = h_3(X') \oplus (ID_i || M || h_i || X)$, $w = h_4(N || U_1 || \kappa_i || T_{us})$, $U_2 = \mathcal{D}_i + xw + \kappa_i$, where T_{us} denotes the current timestamp. U_i sends $\{N, U_1, U_2, w, T_{us}\}$ to S_j via a public channel.

- (2) On receiving the message, S_j first checks whether the timestamp is fresh. If not, S_j rejects the session. Otherwise, S_j computes $X' = e(U_1, \mathcal{D}_{S_j})$ by its secret key and $ID_i || M || h_i || X = N \oplus h_3(X')$. S_j invokes the smart contract to get $\{C_i, \kappa_i P, R_i, t_s, \text{status}\}$ by input parameters h_i . After that, S_j checks whether the equation $U_2 P = R_i + h_i P_{\text{pub}} + wX + \kappa_i P$ holds. If not, S_j fails to authenticate U_i . Otherwise, S_j calculates $R_{U_i S_j} \equiv M \bmod N_{S_j}$ to check whether the access control permission has expired by the equation $R_{U_i S_j} > |t_e - t_s|$, where t_e denotes current time. If expired/not authorized, S_j terminates the request. Otherwise, S_j selects a random number $y \in Z_q^*$, computes $Y = yP$, $K = yX$, $V_j = h_5(ID_i || ID_{S_j} || M || R_{U_i S_j} || X || Y || K || T_{su})$, and sets the session key

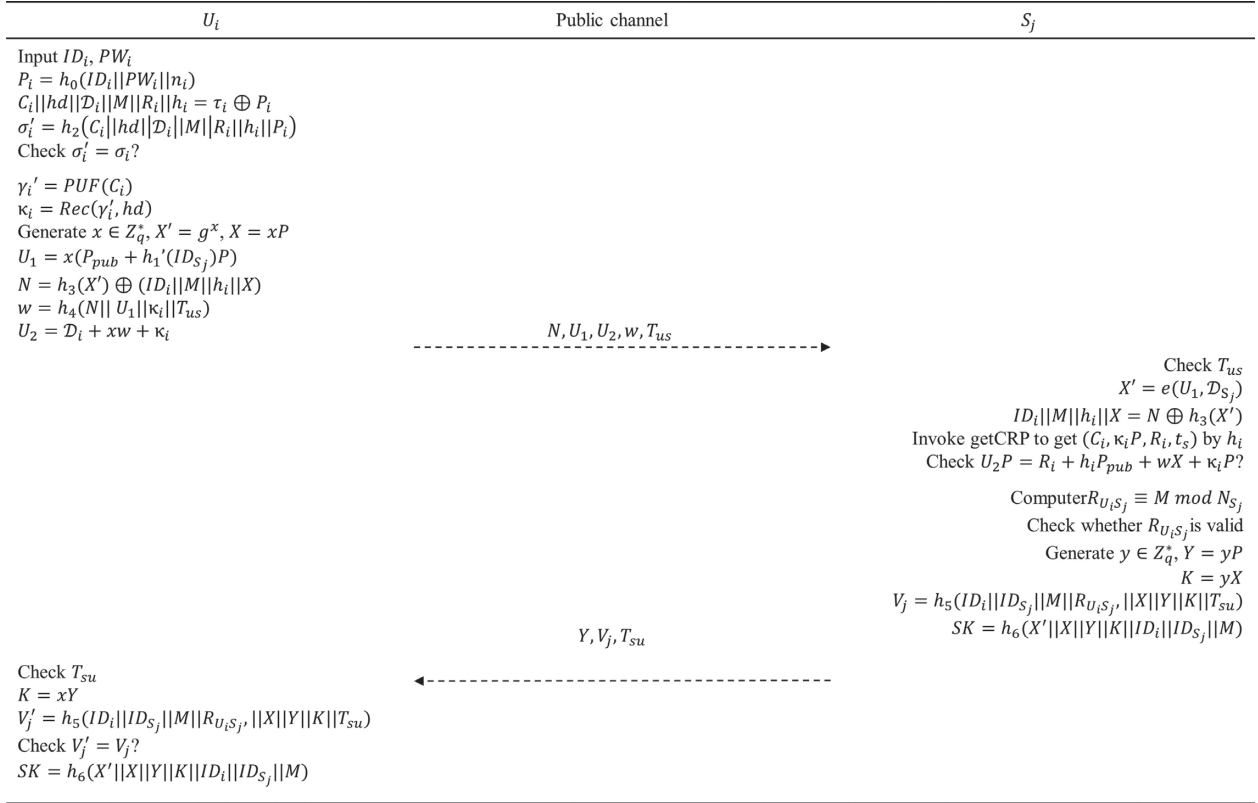


FIGURE 5: Mutual authentication.

$SK = h_6(X' || X || Y || K || ID_i || ID_{S_j} || M)$, where T_{su} is the current timestamp. Then, S_j sends $\{Y, V_j, T_{su}\}$ to U_i .

Note that the server can obtain X' and validate the user identity from the two following equations:

$$\begin{aligned}
 e(U_1, \mathcal{D}_{S_j}) &= e\left(x(P_{pub} + h_1'(ID_{S_j})P), \frac{1}{s + h_1'(ID_{S_j})}P\right) \\
 &= e\left(x(sP + h_1'(ID_{S_j})P), \frac{1}{s + h_1'(ID_{S_j})}P\right) \\
 &= e(P, P)^{x(s + h_1'(ID_{S_j}))} = e(P, P)^x \\
 &= g^x = X' \\
 U_2 P &= (\mathcal{D}_i + xw + \kappa_i)P \\
 &= \mathcal{D}_i P + xwP + \kappa_i P \\
 &= (r_i + sh_i)P + wX + \kappa_i P = R_i + h_i P_{pub} + wX + \kappa_i P.
 \end{aligned} \tag{5}$$

(3) After receiving the message, U_i first checks whether T_{su} is fresh. If not, U_i terminates the session. Otherwise, U_i computes $K = xY$, $V'_j = h_5(ID_i || ID_{S_j} || M || R_{U_i S_j} || X || Y || K || T_{su})$

$\|Y || K || T_{su})$ to check whether V'_j and V_j are equal. If not, U_i aborts the session. Otherwise, U_i calculates the session key $SK = h_6(X' || X || Y || K || ID_i || ID_{S_j} || M)$.

5.6. *Password Update Phase.* If U_i wants to update the password, the following steps are executed between U_i and his/her mobile device:

- (1) U_i inputs ID_i , old PW_i , and new PW_i^* into the mobile device.
- (2) The mobile device computes $P_i = h_0(ID_i \| PW_i \| n_i)$, $C_i \| h d \| \mathcal{D}_i \| M \| R_i \| h_i = \tau_i \oplus P_i$, $\sigma_i = h_2(C_i \| h d \| \mathcal{D}_i \| M \| R_i \| h_i \| P_i)$ and checks whether $\sigma'_i = \sigma_i$ holds. If not, the mobile device rejects the request. Otherwise, the mobile device calculates $P_i^* = h_0(ID_i \| PW_i^* \| n_i)$, $\tau_i^* = (C_i \| h d \| \mathcal{D}_i \| M \| R_i \| h_i) \oplus P_i^*$, $\sigma_i^* = h_2(C_i \| h d \| \mathcal{D}_i \| M \| R_i \| h_i \| P_i^*)$. Then, the mobile device replaces $\{\tau_i, \sigma_i\}$ with $\{\tau_i^*, \sigma_i^*\}$.

5.7. *Access Rights Update Phase.* U_i must update the access control list and send a new ACL_i^* to the RC after the renewal of subscription. The steps will be carried out between U_i and RC.

- (1) U_i generates a new access control list ACL_i^* and sends $\{ID_i, P_i, ACL_i^*\}$ to RC via a secure channel.
- (2) After receiving the message, RC obtains the old access control list ACL_i from the user registration table by the index ID_i and updates ACL_i . Then, RC computes the new common solution M' using CRT. After that, RC regenerates $r'_i \in Z_q^*$, $h'_i = h_1(ID_i \| M' \| R'_i)$, $\mathcal{D}'_i = (r'_i + sh'_i) \bmod q$, $\tau'_i = (C_i \| h d \| \mathcal{D}'_i \| M' \| R'_i \| h'_i) \oplus P_i$, $\sigma'_i = h_2(C_i \| h d \| \mathcal{D}'_i \| M' \| R'_i \| h'_i \| P_i)$ and sends $\{\tau'_i, \sigma'_i\}$ to U_i . Finally, RC updates the user registration table $\{ID_i, P_i, ACL_i^*\}$.

6. Security Analysis

We will demonstrate that the proposed scheme is provably secure under the random oracle model and satisfy the security requirements mentioned in Section 4.3.

6.1. *Security Model.* We define the formal security model for the proposed scheme through a game played between an adversary \mathcal{A} and a challenger \mathcal{C} . Let Π_p^i denote the i^{th} instance of the participant P , where $P \in \{U_i, S_j\}$ denotes two participants: the mobile user and the server who are involved in the execution of the proposed scheme. In the game, \mathcal{A} can initiate a series of queries to \mathcal{C} , which can answer them as follows:

- (i) $h_i(m)$: when \mathcal{A} sends a message m , \mathcal{C} first checks whether m is in the hash-list L_{h_i} . If so, return the value. Otherwise, \mathcal{C} generates a random number $r_i \in Z_q^*$ and stores (m_i, r_i) in the hash-list L_{h_i} . After that, return r_i to \mathcal{A} .
- (ii) *ExtractUserID* (ID_{U_i}): when \mathcal{A} sends a query with the user's U_i identity ID_{U_i} , \mathcal{C} generates a private key and stores (ID_{U_i}, d_i) in the user-list L_{U_i} .

- (iii) *ExtractServerID* (ID_{S_j}): when \mathcal{A} sends a query with the server's S_j identity ID_{S_j} , \mathcal{C} generates a private key and stores (ID_{S_j}, d_j) in the server-list L_{S_j} .
- (iv) *Send* (Π_p^i, m_i): when \mathcal{A} sends a query with a message m_i , \mathcal{C} responds with the result by executing the proposed scheme. \mathcal{A} can start the proposed scheme by sending (Π_p^i, START) .
- (v) *Reveal* (Π_p^i): when \mathcal{A} sends this query, \mathcal{C} returns the session key of the i^{th} instance.
- (vi) *Corrupt* (Π_p): when \mathcal{A} sends this query with the participant $P \in \{U_i, S_j\}$ identity, \mathcal{C} returns the corresponding private key.
- (vii) *Test* (Π_p^i): \mathcal{A} can send this query only once. Upon receiving the query, \mathcal{C} flips a coin $c \in \{0, 1\}$. If $c = 1$, \mathcal{C} returns the session key of i^{th} instance. Otherwise, \mathcal{C} selects a random number and returns it.

After executing the aforementioned queries, \mathcal{A} output the guess c' in the test query phase. If $c' = c$, we say \mathcal{A} breaks the semantic security of the proposed scheme. The advantage that \mathcal{A} violates the authenticated key agreement (AKA) of this scheme Γ is denoted by $A \text{adv}_{\Gamma}^{\text{AKA}} = |2\text{Pr}[c' = c] - 1|$.

Definition 1. (AKA-security): we say an authentication scheme Γ is AKA-security if $A \text{adv}_{\Gamma}^{\text{AKA}} = |2\text{Pr}[c' = c] - 1|$ is negligible for any polynomial adversary \mathcal{A} .

Let E_{U-S} and E_{S-U} denote the events \mathcal{A} that can violate $U - to - S$ authentication by forging a login message and $S - to - U$ authentication by forging a response message, respectively. The advantage that \mathcal{A} violates the mutual authentication of this scheme Γ is denoted by $A \text{adv}_{\Gamma}^{\text{MA}} = |\text{Pr}[E_{U-S}] + \text{Pr}[E_{S-U}]|$.

Definition 2. (MA-security): we say an authentication scheme Γ is MA-security if $A \text{adv}_{\Gamma}^{\text{MA}} = \text{Pr}[E_{U-S}] + \text{Pr}[E_{S-U}]$ is negligible for any polynomial adversary \mathcal{A} .

6.2. *Provable Security.* We will prove that our proposed scheme is MA-security and AKA-security in the security model above.

Lemma 1. *No polynomial adversary can forge a legal login message if the DL problem is difficult.*

Proof. suppose the adversary \mathcal{A} can output a legal login message with non-negligible probability ϵ , then the challenger \mathcal{C} can solve the DL problem with non-negligible advantage.

Given a DL instance $(P, \theta = \tau P)$, the task of \mathcal{C} is to compute τ , where $\tau \in Z_q^*$ is unknown to \mathcal{C} . \mathcal{C} generates a random number $\hat{\tau} \in Z_q^*$, sets $P_{\text{pub}} \leftarrow \theta$, $\hat{P}_{\text{pub}} = \hat{\tau} P$, and sends the system parameters $\text{params} = \{G_1, G_2, P, e, q, P_{\text{pub}}, \hat{P}_{\text{pub}}, h_0, h_1, h'_1, h_2, h_3, h_4, h_5, h_6\}$ to \mathcal{A} . \mathcal{C} will maintain seven hash-lists L_{h_i} ($i = 0, 1, 2, 3, 4, 5, 6$), a mobile user list L_{U_i} , and a server list L_{S_j} . \mathcal{C} randomly chooses a user's identity $ID_{U_i^*}$ as the challenge identity and answers \mathcal{A} 's queries as follows:

- (i) $h_i(m)$: \mathcal{C} maintains the lists L_{h_i} , which are initially empty. \mathcal{C} , firstly, checks whether (m_i, r_i) exists in L_{h_i} . If it does, \mathcal{C} returns r_i to \mathcal{A} . Otherwise, \mathcal{C} generates a random number r_i , stores (m_i, r_i) in L_{h_i} , and returns r_i to \mathcal{A} .
- (ii) *ExtractUserID* (ID_{U_i}): \mathcal{C} maintains the mobile user list L_{U_i} , which is initially empty. \mathcal{C} , firstly, checks whether $(ID_{U_i}, R_i, d_i, \kappa_i)$ exists in L_{U_i} . If it does, \mathcal{C} returns ID_{U_i} to \mathcal{A} . Otherwise, \mathcal{C} executes the following steps:
- (a) If $ID_{U_i} \neq ID_{U_i^*}$, \mathcal{C} randomly selects $r_{U_i}, h_{1_{U_i}} \in Z_q^*$, and a random number κ_i as the output of fuzzy extractor. \mathcal{C} computes $R_{U_i} = r_{U_i}P - h_{1_{U_i}}P_{pub}$. Otherwise, \mathcal{C} sets $d_i \leftarrow r_{U_i}$ and stores $(ID_{U_i}, R_i, d_i, \kappa_i)$ and $(ID_{U_i}, R_{U_i}, h_{1_{U_i}})$ in L_{U_i} and L_{h_i} , respectively, and returns ID_{U_i} to \mathcal{A} .
- (b) If $ID_{U_i} = ID_{U_i^*}$, \mathcal{C} generates two random numbers $r_{U_i}^*, h_{1_{U_i}}^* \in Z_q^*$, computes $R_{U_i}^* = r_{U_i}^*P$, sets $d_{U_i}^* \leftarrow \perp$, selects a random number κ^* as the output of fuzzy extractor, stores $(ID_{U_i^*}, R_{U_i}^*, \perp, \kappa^*)$ and $(ID_{U_i^*}, R_{U_i}^*, h_{1_{U_i}}^*)$ in L_{U_i} and L_{h_i} , respectively, and returns $ID_{U_i^*}$ to \mathcal{A} .
- (iii) *ExtractServerID* (ID_{S_j}): \mathcal{C} maintains the server list L_{S_j} , which is initially empty. \mathcal{C} , firstly, checks whether (ID_{S_j}, d_j) is in L_{S_j} . If it does, \mathcal{C} returns ID_{S_j} to \mathcal{A} . Otherwise, \mathcal{C} selects $e_j \in Z_q^*$, computes $d_j = 1/\tau + e_jP$, stores (ID_{S_j}, d_j) and (ID_{S_j}, e_j) in L_{S_j} and L_{h_i} , respectively, and then returns ID_{S_j} to \mathcal{A} .
- (iv) *Send* (Π_p^i, m_i): \mathcal{A} can send the following queries:
- (a) *Send* (U_i^t, START): when \mathcal{A} sends this query, \mathcal{C} , firstly, checks if $ID_{U_i} = ID_{U_i^*}$ holds. If it does, \mathcal{C} aborts the game. Otherwise, \mathcal{C} checks whether ID_{U_i} exists in L_{U_i} . If not, \mathcal{C} executes the *ExtractUserID* (ID_{U_i}) query. After that, with the private key d_i , \mathcal{C} generates a random number $x \in Z_q^*$ and calculates $X' = g^x, X = xP, U_1, U_2, N, \omega$ as described. \mathcal{C} stores $(ID_{U_i}, ID_{S_j}, t, x, X, M, R_{U_i, S_j})$ in L_{U_i, S_j} and returns (U_1, U_2, N, ω) to \mathcal{A} .
- (b) *Send* ($S_j^t, N, U_1, U_2, \omega, T_{us}$): when \mathcal{A} sends this query, \mathcal{C} checks whether ID_{S_j} exists in L_{S_j} . If not, \mathcal{C} executes the *ExtractServerID* (ID_{S_j}). After that, with the private key d_j , \mathcal{C} computes $X' = e(U_1, d_j)$ and extracts $ID_{U_i} \| M \| h_i \| X = N \oplus h_3(X')$. \mathcal{C} obtains h_i, κ_i from L_{U_i} , verifies if $U_2P = R_i + h_iP_{pub} + \omega X + \kappa_iP$ holds. If not, \mathcal{C} rejects the message. Otherwise, \mathcal{C} generates a random number $y \in Z_q^*$, computes Y, V_j, T_{su} as described, and returns it to \mathcal{A} . If $ID_{U_i} = ID_{U_i^*}$, then \mathcal{A} successfully forges a legal login message.
- (c) *Send* (U_i^t, Y, V_j, T_{su}): upon receiving this message, \mathcal{C} checks if $V_j' = V_j$ holds with $(ID_{U_i}, ID_{S_j}, t, x, X, M, R_{U_i, S_j})$ in L_{U_i, S_j} . If not, \mathcal{C} rejects the message. Otherwise, \mathcal{C} authenticates \mathcal{A} .
- (v) *Reveal* (Π_p^i): upon receiving this message, \mathcal{C} returns the session key SK if SK is accepted. Otherwise, \mathcal{C} returns ' \perp ' to \mathcal{A} .
- (vi) *Corrupt* (Π_p): upon receiving the identity ID_{U_i} (or ID_{S_j}), \mathcal{C} looks up L_{U_i} (or L_{S_j}) and returns d_i (or d_j) to \mathcal{A} .
- (vii) *Test* (Π_p^i): \mathcal{C} generates a random number with the same length of session key and returns it to \mathcal{A} .

Suppose \mathcal{A} forges a legal login message with $ID_{U_i} = ID_{U_i^*}$. By applying forking lemma, \mathcal{A} can generate another legal login message U_2 with the same input of the simulation and different hash oracle queries. Then, we can get the following equations:

$$U_2P = R_i^* + h_i^*P_{pub} + \omega X + \kappa_i^*P, \quad (6)$$

$$U_2'P = R_i^* + h_i^*P_{pub} + \omega X + \kappa_i^*P. \quad (7)$$

Based on equations (7) and (8), we have the following:

$$\begin{aligned} (U_2 - U_2')P &= (R_i^* + h_i^*P_{pub} + \omega X + \kappa_i^*P) \\ &\quad - (R_i^* + h_i^*P_{pub} + \omega X + \kappa_i^*P) \\ &= (h_i^* - h_i^*)P_{pub} \\ &= (h_i^* - h_i^*)\tau P. \end{aligned} \quad (8)$$

By equations (8), $(U_2 - U_2')(h_i^* - h_i^*)^{-1} \bmod q$ is the answer of DL problem. The advantage that \mathcal{C} solves the DL problem is given below. Firstly, some events are defined as follows:

- (i) E_1 : the simulation does not abort in any *Send* query.
(ii) E_2 : \mathcal{A} successfully forges a legal message.
(iii) E_3 : $ID_{U_i} = ID_{U_i^*}$.

Let q_{h_1} and q_s denote the number of h_1 queries and *Send* queries. Then, we have the following equations:

$$\Pr[E_1] = \left(1 - \frac{1}{q_s + 1}\right)^{q_s}$$

$$\Pr[E_2|E_1] \geq \epsilon \quad (9)$$

$$\Pr[E_3|E_2 \wedge E_1] \geq \frac{1}{q_{h_1}}$$

By equation (9), the advantage that \mathcal{C} solves the DL problem is as follows:

$$\begin{aligned} \epsilon &= \Pr[E_3 \wedge E_2 \wedge E_1] \\ &= \Pr[E_3|E_2 \wedge E_1] \cdot \Pr[E_2|E_1] \cdot \Pr[E_1] \geq \left(1 - \frac{1}{q_s + 1}\right)^{q_s} \frac{1}{q_{h_1}} \cdot \epsilon. \end{aligned} \quad (10)$$

Thus, \mathcal{C} can solve the DL problem with non-negligible probability ϵ by playing the game with \mathcal{A} . It contradicts with the difficulty of DL problem. We conclude that no

polynomial adversary can forge a legal login message with non-negligible probability. \square

Lemma 2. *No polynomial adversary can forge a legal response message if the k -mBIDH Problem is difficult.*

Proof. suppose the adversary \mathcal{A} can forge a legal response message with non-negligible probability ε , then the challenger \mathcal{C} can solve the k -mBIDH problem with non-negligible advantage. Given a k -mBIDH instance, $P, P_{\text{pub}} = sP, \iota P, \{e_1, e_2, \dots, e_k\} \in Z_q^*, 1/s + e_1P, 1/s + e_2P, \dots, 1/s + e_kP \in G_1$, the task of \mathcal{C} is to compute $e(P, P)^{\iota/s + e^*}$, where s, ι are unknown to \mathcal{C} and $e^* \notin \{e_1, e_2, \dots, e_k\}$. \mathcal{C} sends the system parameters $\text{params} = \{G_1, G_2, q, e, g, P, P_{\text{pub}}, h_0, h_1, h'_1, h_2, h_3, h_4, h_5, h_6\}$ to \mathcal{A} . \mathcal{C} will maintain seven hash-lists L_{h_i} ($i=0, 1, 2, 3, 4, 5, 6$), a mobile user list L_{U_i} , and a server list L_{S_j} . \mathcal{C} chooses a random identity $ID_{S_j^*}$ as the challenge identity.

h_i ($i=0, 1, 2, 3, 4, 6$), Reveal, Corrupt, and Test query are the same as those in the simulation of Lemma 1. h_5 , Extract, and Send query are executed as follows:

- (i) *ExtractServerID* (ID_{S_j}): \mathcal{C} maintains the server list L_{S_j} , which is initially empty. \mathcal{C} , firstly, checks whether (ID_{S_j}, e_j) is in L_{S_j} . If it is, \mathcal{C} returns ID_{S_j} to \mathcal{A} . Otherwise, \mathcal{C} executes the following steps:
 - (a) If $ID_{S_j} \neq ID_{S_j^*}$, \mathcal{C} selects $e_j \in \{e_1, e_2, \dots, e_k\}$ and $d_j = (1/s + e_j)P$. \mathcal{C} stores (ID_{S_j}, d_j) and (ID_{S_j}, e_j) in L_{S_j} and $L_{h'_1}$, respectively, and returns ID_{S_j} to \mathcal{A} .
 - (b) If $ID_{S_j} = ID_{S_j^*}$, \mathcal{C} sets $h'_1(ID_{S_j}) = e^*$, stores (ID_{S_j}, \perp) and (ID_{S_j}, e^*) in L_{S_j} and $L_{h'_1}$, respectively, and returns ID_{S_j} to \mathcal{A} .
- (ii) *ExtractUserID* (ID_{U_i}): \mathcal{C} maintains the mobile user list L_{U_i} , which is initially empty. \mathcal{C} , firstly, checks whether $(ID_{U_i}, R_{U_i}, d_i, \kappa_i)$ exists in L_{U_i} . If it does, \mathcal{C} returns ID_{U_i} to \mathcal{A} . Otherwise, \mathcal{C} randomly selects $r_{U_i}, h_{1_{U_i}} \in Z_q^*$ and computes $R_{U_i} = r_{U_i}P - h_{1_{U_i}}P_{\text{pub}}$. \mathcal{C} sets $d_i = r_{U_i}$ and generates a random number κ_i as the output of fuzzy extractor. \mathcal{C} stores $(ID_{U_i}, R_{U_i}, d_i, \kappa_i)$ and $(ID_{U_i}, R_{U_i}, h_{1_{U_i}})$ in L_{U_i} and L_{h_1} , respectively, and returns ID_{U_i} to \mathcal{A} .
- (iii) *Send* (Π_p^i, m_i): \mathcal{A} can send the following queries:
 - (a) *Send* (U_i^t, START): If U_i 's partner is S_j^* , \mathcal{C} sets $U_1 = \iota P$ and calculates U_2, N, ω as described. Otherwise, \mathcal{C} looks up $(ID_{U_i}, R_{U_i}, h_{1_{U_i}})$ from L_{h_1} and generates a legal login message as described.
 - (b) *Send* ($S_j^t, N, U_1, U_2, \omega, T_{su}$): When \mathcal{A} sends this query, \mathcal{C} checks whether $ID_{S_j} = ID_{S_j^*}$ holds. If it does, \mathcal{C} aborts the game. Otherwise, \mathcal{C} behaves the operations as described.
 - (c) *Send* (U_i^t, Y, V_j, T_{su}): Upon receiving this message, \mathcal{C} checks if $V^t = V_j$ holds.

If not, \mathcal{C} rejects the message. Otherwise, \mathcal{C} authenticates \mathcal{A} . If $ID_{S_j} = ID_{S_j^*}$, it means that \mathcal{A} successfully forges a legal response message.

Suppose \mathcal{A} forges a legal response message with $ID_{S_j} = ID_{S_j^*}$. In the game, \mathcal{A} must send $V_j = h_5(ID_{U_i}, ID_{S_j}, M, R_{U_i}, X, Y, K, T_{su})$ query after recovering $ID_{U_i} \| M \| h_{1_{U_i}} \| X = N \oplus h_3(X')$. Thus, \mathcal{A} must have executed the h_3 query with the message X' . We have the following:

$$\begin{aligned} X' &= e(U_1, d_j^*) \\ &= e\left(\iota P, \frac{1}{s + e^*} P\right) \\ &= e(P, P)^{\iota/s + e^*} \end{aligned} \quad (11)$$

The advantage that \mathcal{C} solves the k -mBIDH Problem is given below. Some events are defined as follows:

- (i) E_1 : the simulation does not abort.
- (ii) E_2 : \mathcal{A} successfully forges a legal response message.
- (iii) E_3 : $ID_{S_j} = ID_{S_j^*}$.
- (iv) E_4 : \mathcal{C} selects a correct tuple from L_{h_3} .

Let q_{h_1}, q_{h_3} , and q_s denote the number of h_1, h_3 query and Send query. Then, we have the following equations:

$$\begin{aligned} \Pr[E_1] &= \left(1 - \frac{1}{q_s + 1}\right)^{q_s} \\ \Pr[E_2|E_1] &\geq \varepsilon \\ \Pr[E_3|E_2 \wedge E_1] &\geq \frac{1}{q_{h_1}} \\ \Pr[E_4|E_3 \wedge E_2 \wedge E_1] &\geq \frac{1}{q_{h_3}} \end{aligned} \quad (12)$$

By equation (12), the advantage that \mathcal{C} solves the k -mBIDH Problem is as follows:

$$\begin{aligned} \varepsilon &= \Pr[E_4 \wedge E_3 \wedge E_2 \wedge E_1] \\ &= \Pr[E_4|E_3 \wedge E_2 \wedge E_1] \cdot \Pr[E_3|E_2 \wedge E_1] \cdot \Pr[E_2|E_1] \cdot \Pr[E_1] \\ &\geq \left(1 - \frac{1}{q_s + 1}\right)^{q_s} \frac{1}{q_{h_1}} \frac{1}{q_{h_3}} \cdot \varepsilon. \end{aligned} \quad (13)$$

Thus, \mathcal{C} can solve the k -mBIDH problem with non-negligible probability ε by playing the game with \mathcal{A} . It contradicts with the hardness of k -mBIDH problem. We conclude that no polynomial adversary can forge a legal response message with non-negligible probability. \square

Theorem 1. *The proposed scheme is MA-security if the DL problem and k -mBIDH problem are hard.*

Proof. Lemma 1 and 2 demonstrate that no polynomial adversary can forge a legal login message or response message if the DL problem and k -mBIDH problem are hard. In other words, the mobile user U_i and the server S_j can

authenticate each other. Therefore, the proposed scheme is MA-security. \square

Theorem 2. *The proposed scheme is AKA-security if CDH problem is hard.*

Proof. suppose \mathcal{A} can guess b correctly with non-negligible probability ε in the Test query, then \mathcal{C} can solve the CDH problem with non-negligible probability. Some events are defined as follows:

- (i) E_b : \mathcal{A} guesses the value of b correctly.
- (ii) E_{TU} : \mathcal{A} makes the Test query to the user.
- (iii) E_{TS} : \mathcal{A} makes the Test query to the server.
- (iv) E_{US} : \mathcal{A} forges a legal login message between the user and server.

Since the probability that \mathcal{A} can guess the value of b correctly is at least $1/2$, then we have $\Pr[E_b] \geq \varepsilon/2$. We can get the equations as follows:

$$\begin{aligned} \frac{\varepsilon}{2} &\leq \Pr[E_b] = \Pr[E_b \wedge E_{TU}] \\ &\quad + \Pr[E_b \wedge E_{TS} \wedge E_{US}] \\ &\quad + \Pr[E_b \wedge E_{TS} \wedge E_{US}] \\ &\leq \Pr[E_b \wedge E_{TU}] + \Pr[E_{US}] \\ &\quad + \Pr[E_b \wedge E_{TS} \wedge E_{US}]. \end{aligned} \quad (14)$$

Then,

$$\Pr[E_b \wedge E_{TU}] + \Pr[E_b \wedge E_{TS} \wedge E_{US}] \geq \frac{\varepsilon}{2} - \Pr[E_{US}]. \quad (15)$$

The event $E_{TS} \wedge E_{US}$ and E_{TU} are equal. We can get the following:

$$2\Pr[E_b \wedge E_{TU}] \geq \frac{\varepsilon}{2} - \Pr[E_{US}]. \quad (16)$$

Then, the advantage that \mathcal{C} solves the CDH problem is as follows:

$$\epsilon = \Pr[E_b \wedge E_{TU}] \geq \frac{\varepsilon}{4} - \frac{\Pr[E_{US}]}{2}. \quad (17)$$

The event $E_b \wedge E_{TU}$ means that \mathcal{A} impersonates the user and has $K = xY$, which is the solution of CDH problem. According to Lemma 1, $\Pr[E_{US}]$ is negligible. Then, we can get $\Pr[E_b \wedge E_{TU}]$, which is non-negligible because ε is non-negligible. It means that \mathcal{C} can solve the CDH problem with non-negligible probability ϵ by playing the game with \mathcal{A} . It contradicts the difficulty of CDH problem. We conclude that no polynomial adversary can guess b correctly, and the proposed scheme is AKA-security. \square

6.3. Security Requirement Analysis. We also show that the proposed scheme satisfies the security requirements described in Section 4.3.

Single registration: According to the description of the proposed scheme, the trusted RC generates the registration information for users. Then, users can be authenticated by other healthcare service providers.

No online registration center: According to the specification of the proposed scheme, RC is not involved in the mutual authentication.

Mutual authentication: According to the proof of Lemma 1 and Lemma 2, we know that no polynomial adversary can forge a legal login message and response message. Thus, the user U_i and the server S_j can authenticate each other by the received message.

User anonymity: Based on the description of the proposed scheme, the user identity ID_{U_i} is hidden in the login message $\{N, U_1, U_2, \omega, T_{us}\}$, where $N = h_3(X') \oplus (ID_{U_i} \| M \| h_i \| X)$. An adversary can extract ID_{U_i} only if he/she can get $X' = e(U_1, \mathcal{D}_{S_j})$ after solving the k -mBIDH problem. Besides, the registration recorded in the blockchain is hidden by the hash function $h_i = h_1(ID_i \| M \| R_i)$.

Untraceability: In the proposed scheme, random x is generated in each new session to compute a new login message $\{N, U_1, U_2, \omega, T_{us}\}$, where $X' = g^x$, $X = xP$, $U_1 = x(PK + h_1(ID_{S_j})P)$, $N = h_3(X') \oplus (ID_{U_i} \| M \| h_i \| X)$, $\omega = h_4(N, U_1, \kappa_i, T_{us})$, $U_2 = \mathcal{D}_{U_i} + xw + \kappa_i$. Because of the randomness of x , the adversary cannot find any relation among these login messages of different sessions and thus cannot trace the users' behavior.

Session key agreement: Based on the description of the proposed scheme, the user and the server will generate a session key $SK = h_6(X' \| X \| Y \| K_{ij} \| ID_i \| ID_{S_j})$ for future secure communications. We know that no adversary can compute xyP from xP and yP since the CDH problem is hard.

Perfect forward secrecy: Assume that there is an adversary who gets the long-term private key of the user and the server and intercepts the exchange message of previous sessions. Then, the adversary cannot compute $K_{ij} = xyP$ in $SK = h_6(X' \| X \| Y \| K_{ij} \| ID_i \| ID_{S_j})$ even if he gets X and Y from U_1, N, Y, V_j since the CDH problem is hard. Therefore, the adversary cannot generate previous session keys even if he knows both private keys of the user and server.

Two-factor security with PUF: Assume that an adversary steals the mobile device and extracts the data $n_i, P_i, \tau_i, \sigma_i$ by a side channel attack. He can guess the password PW_i , however, he cannot verify the correctness of the password without knowing the identity ID_{U_i} . He cannot get the response output of PUF without C_i .

Moreover, the adversary cannot be authenticated in his mobile device because of the uncloneability of PUF even if he knows the password. Thus, our proposed scheme can satisfy the two-factor security.

Access control for data privacy: The proposed scheme provides access control in the authentication process. The common solution M can be used to determine the access permission $\text{Num}R_i$. The server can determine the access permission of the user efficiently without any other access control list to make data disclosure minimum. Besides, the common solution is associated with the RC's private key s , which means no adversary can forge the access permission message.

Resistance of various attacks: To resist known attacks existing in the service system, the proposed scheme should resist known attacks.

- (i) *Insider attack:* An insider in the system only knows user identity ID_i but cannot verify the password or private key without n_i, P_i . Thus, the proposed scheme can resist an insider attack.
- (ii) *Stolen card attack:* An adversary gets the user's smart card and extracts the data $n_i, P_i, \tau_i, \sigma_i$. However, he cannot verify the password without the user's identity. Therefore, the proposed scheme can withstand the stolen card attack.
- (iii) *Offline password guessing attack:* An adversary gets the user's smart card and extracts the data $n_i, P_i, \tau_i, \sigma_i$. However, he cannot verify the password without the user's identity. Therefore, the proposed scheme can resist the offline password guessing attack.
- (iv) *Replay attack:* According to the description of the proposed scheme, we use the timestamp to check the freshness. Besides, the user and the server will generate new random numbers $x, y \in Z_q^*$ in each session. Both of them can find the replay of a message by the validity of the received message.
- (v) *User impersonation attack:* Based on the proof of Lemma 1, we show that no adversary can forge a legal login message without the user's private key. The server can check the validity of the login message.
- (vi) *Server impersonation attack:* Based on the proof of Lemma 2, we show that no adversary can forge a legal response message without server's private key. The user can check the validity of the response message V_j since the server will extract X', X using its private key.
- (vii) *Man-in-the-middle attack:* According to Theorem 1, we conclude that our proposed scheme provides mutual authentication, which means no adversary can forge a legal message without knowing the private keys.
- (viii) *Physical attack:* No adversary can recreate the same PUF to authenticate the device since PUFs are almost impossible to clone. Besides, the response R and helper data hd are kept a secret, and

only (C_i, g^k) is stored in the blockchain, which can resist the modeling attack.

6.4. Security Comparisons. We now compare our proposed scheme with other prior related schemes, namely those of Xiong et al. [34], Jia et al. [21], and Son et al. [6]. As shown in Table 4, [6, 21, 34] cannot provide two-factor security, in which PUF serves as one of the authentication factors. [21] cannot support access control in the authentication process. All of them cannot resist physical attacks and others. Therefore, our scheme can satisfy all the listed security requirements and have better security.

7. Performance Analysis

In this section, we discuss the computational and communication costs of the proposed scheme and other related schemes [6, 21, 34, 37].

We will directly use the parameters of Jia et al. [21] scheme for the comparison in Table 5. We do not consider the execution time of registration phases, since they are executed only once. We also focus on the communication costs in the registration phase. The size of the element in G_1, G_2 , and Z_q is 1024, 1024, and 160 bits, respectively, denoted by $|G_1|, |G_2|, |Z_q|$. Suppose the output length of the hash function $|H|$ and user's identity $|ID|$ is 256 bits. The length of the timestamp $|T|$ and common solution is 32 bits.

The summary of computation and communication costs is shown in Tables 6, 7 and Figures 6, 7. The proposed scheme has a lower computation cost compared to that in the scheme [6, 34] on the mobile users' side and [6, 21] on the server side. Besides, the proposed protocol has lower communication costs compared to those in [21]. Though the scheme [6, 34] has lower communication costs than ours, our scheme can be more efficient and meet more desirable security requirements.

We implement a software prototype based on Hyperledger Fabric 2.0 on a single machine running an Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz, 8 GB RAM, and Ubuntu 18.04 for 64 bits operations system. We deploy a test network that consists of three peer organizations and a single orderer organization with three ordering nodes in a single channel to simulate the healthcare provider servicers in certain areas. The block size is 100 transactions in a batch and the block timeout is 2s to wait before creating a batch. The raft consensus algorithm is used for agreement on the consistence of transactions across the network. The smart contracts written in Go are deployed over the network, which is shown in Algorithm 1 and 2. All transactions of uploading, query, and update are driven via Hyperledger Fabric gateway to evaluate the throughput (TPS) and latency as shown in Figures 8 and 9. The performance demonstrates that the maximum throughput is 310 TPS for uploading the transactions and 600 TPS for query and update. The average latency for all transactions increases as increased in the total transactions. There are lots of factors that will affect the performance of the framework, such as the choices of ledger database, endorsement policy, network configuration parameters, and so on.

```

function uploadSubs( $h_i, ID_{S_j}, R_{U_i S_j}$ ) {
   $ID_{ij} = h_i \parallel ID_{S_j}$ 
  if userExists ( $ID_{ij}$ ) == false
    expiretime = time.Now().Month() +  $R_{U_i S_j}$ 
    service = Service \{ $ID_{ij}, R_{U_i S_j}, expiretime$ \}
    return putState( $ID_{ij}, service$ )
  else
    return Errorf("the user has already exists")
function querySubs( $h_i, ID_{S_j}$ ) {
   $ID_{ij} = h_i \parallel ID_{S_j}$ 
  err, result = getState ( $ID_{ij}$ )
  if (err == null)
    return result
  else
    return err
function updateSubs( $h_i, R_{U_i S_j}$ ) {
   $ID_{ij} = h_i \parallel ID_{S_j}$ 
  if userExists ( $ID_{ij}$ ) == true
    expiretime = time.Now().Month() +  $R_{U_i S_j}$ 
    service = Service \{ $ID_{ij}, R_{U_i S_j}, expiretime$ \}
    return putState( $ID_{ij}, R_{U_i S_j}$ )
  else
    return Errorf("the user does not exist")

```

ALGORITHM 1: Subscription service contract.

```

function uploadCRP( $h_i, C_i, \kappa_i P, R_i, t_s$ ) {
  if userExists ( $h_i$ ) == false
    status = 'valid'
     $t_s = time.Now()$ 
    user = User \{ $h_i, C_i, \kappa_i P, R_i, t_s, status$ \}
    return putState( $h_i, user$ )
  else
    return Errorf("the user has already exists")
function queryCRP ( $h_i$ ) {
  err, result = getState ( $h_i$ )
  if (err == null && result.getStatus() == 'valid')
    return result
  else if (result.getStatus() != "valid")
    return Errorf("the CRP of user has been expired")
  return err
function updateCRP( $h_i, C_i, \kappa_i P, R_i$ ) {
  if userExists ( $h_i$ ) == true
    status = 'valid'
     $t_s = time.Now()$ 
    user = User \{ $h_i, C_i, \kappa_i P, R_i, t_s, status$ \}
    return putState( $h_i, user$ )
  else
    return Errorf("the user does not exist")

```

ALGORITHM 2: Challenge-response pair contract.

TABLE 4: Security comparison.

Security requirements	[34]	[21]	[6]	Ours
Single registration	✓	✓	✓	✓
No online registration center	✓	✓	✓	✓
Mutual authentication	✓	✓	✓	✓
User anonymity	✓	✓	✓	✓
Untraceability	✓	✓	✓	✓
Session key agreement	✓	✓	✓	✓
Perfect forward secrecy	✓	✓	✓	✓
Two-factor security with PUF	×	×	×	✓
Access control	✓	×	✓	✓
Resistance of known attacks	×	×	×	✓

TABLE 5: Running time of basic operation.

Description	Alibaba cloud	Google nexus
T_{G_b} Bilinear pairing	5.275	48.66
T_{G_m} Scalar multiplication	1.97	19.919
T_a Point multiplication	0.012	0.118
T_h Hash function	0.009	0.089
T_e Modular exponentiation	0.339	3.328

TABLE 6: Computation costs comparison (ms).

Scheme	Computation costs(User)	Computation costs(Server)
[34]	$2 T_{G_m} + T_{G_a} + T_e + T_{G_b} + 8 T_{G_h} \approx 92.656$	$2 T_{G_m} + T_{G_a} + T_{G_b} + 5 T_h \approx 9.272$
[21]	$4 T_{G_m} + T_e + 5 T_h \approx 83.449$	$T_b + 5 T_{G_m} + 3 T_{G_a} + 5 T_h \approx 15.206$
[6]	$5 T_{G_m} + 8 T_h \approx 100.307$	$2 T_{G_b} + 5 T_{G_m} + 5 T_h \approx 20.445$
Ours	$4 T_{G_m} + T_e + 7 T_h + T_a \approx 83.745$	$T_{G_b} + 4 T_{G_m} + 3 T_h + 3 T_a \approx 13.218$

TABLE 7: Communication costs comparison.

Scheme	Communication costs	Length/bits
[34]	$2 G + 3 H $	2816
[21]	$4 G + 2 Z_q + 2 H + I D $	4736
[6]	$3 G + 2 H + 2 T $	3648
Ours	$3 G + Z_q + 3 H + I D + 2 T $	4320

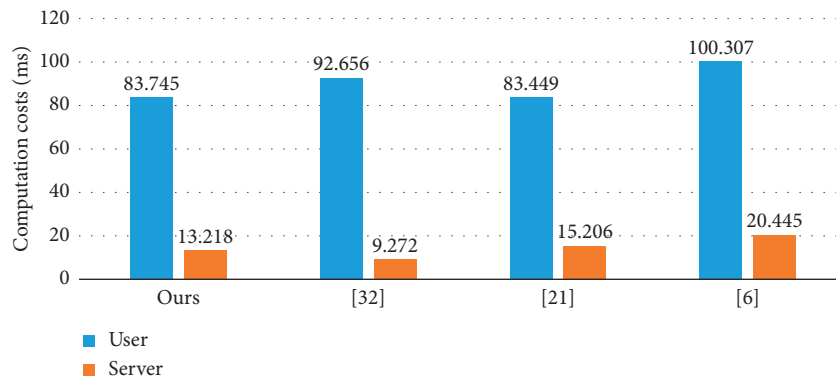


FIGURE 6: Computation costs comparison.

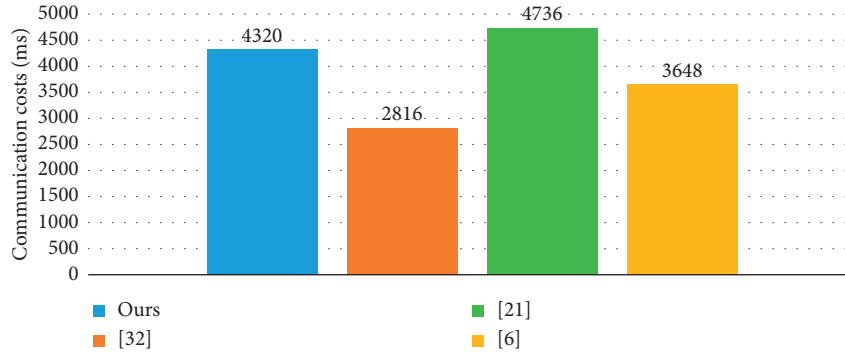


FIGURE 7: Communication costs comparison.

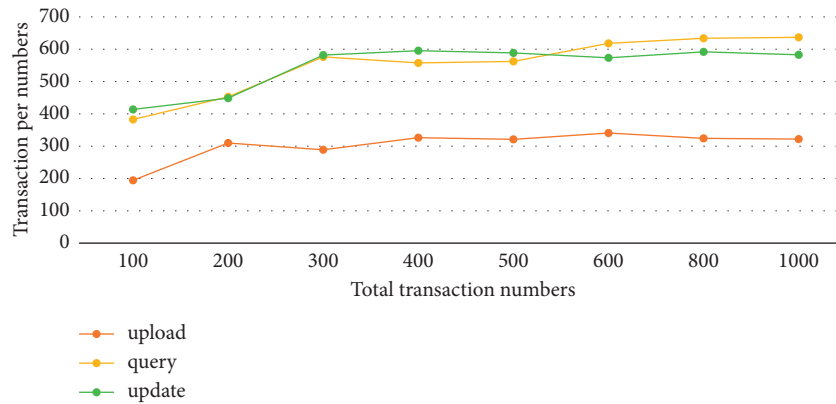


FIGURE 8: Transactions per second in the blockchain.

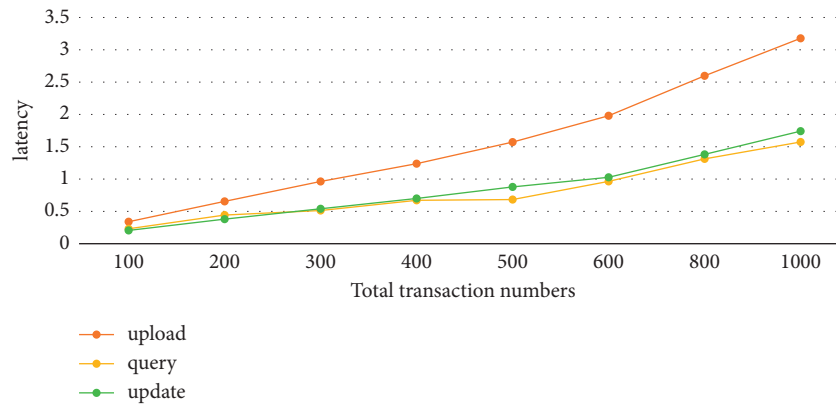


FIGURE 9: Latency for all transactions in the blockchain.

8. Conclusion

In this paper, we proposed a user authentication scheme with access control based on CRT for multiserver architectures, which not only secures sensitive health data transmission but enhances data privacy. The highlight of our protocol is that the decent integration of blockchain and PUF can achieve secure data sharing across medical institutions and identify each device for further security in the telehealth environment. Security analysis shows that our proposed scheme meets all of the desirable requirements above. Moreover, the performance analysis demonstrated

that this protocol has lower communication and computation costs.

Future research will focus on more flexible access control updates and the improvement of computational and communication efficiency for resource-limited devices. We will also explore a privacy-preserving data sharing mechanism based on the blockchain and the improvement of blockchain performance. In addition, there is no doubt that a centralized registration center is convenient and available but our protocol is susceptible to some common security flaws, such as single point of failure, distributed denial of service (DDoS) attacks, and register center compromised attacks.

Decentralized blockchain technology to mitigate central registration center problems is also a significant direction.

Data Availability

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

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The work was supported by the National Key Research and Development Program of China (No. 2021YFA1000600), the Shandong Provincial Key Research and Development Program (Nos. 2021CXGC010107 and 2020CXGC010107), the National Natural Science Foundation of China (Nos. 62172307, U21A20466, 61972294, and 61932016), the Blockchain Core Technology Strategic Research Program of Ministry of Education of China (No. 2020KJ010301), the Special Project on Science and Technology Program of Hubei Province (No. 2020AEA013), the Natural Science Foundation of Hubei Province (No. 2020CFA052), the Wuhan Municipal Science and Technology Project (No. 2020010601012187), and the Guangxi Key Laboratory of Trusted Software (No. kx202001).

References

- [1] Q. Jiang, J. Ma, G. Li, and L. Yang, "An efficient ticket based authentication protocol with unlinkability for wireless access networks," *Wireless Personal Communications*, vol. 77, no. 2, pp. 1489–1506, 2014.
- [2] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "Psspr: a source location privacy protection scheme based on sector phantom routing in wsns," *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1204–1221, 2021.
- [3] L. H. Li-Hua Li, L.-C. Luon-Chang Lin, and M.-S. Min-Shiang Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.
- [4] P. Mohit, R. Amin, A. Karati, G. P. Biswas, and M. K. Khan, "A standard mutual authentication protocol for cloud computing based health care system," *Journal of Medical Systems*, vol. 41, no. 4, p. 50, 2017.
- [5] V. Kumar, S. Jangirala, and M. Ahmad, "An efficient mutual authentication framework for healthcare system in cloud computing," *Journal of Medical Systems*, vol. 42, no. 8, pp. 1–25, 2018.
- [6] S. Son, J. Lee, M. Kim, S. Yu, A. K. Das, and Y. Park, "Design of secure authentication protocol for cloud-assisted telecare medical information system using blockchain," *IEEE Access*, vol. 8, Article ID 192177, 2020.
- [7] M. A. Khan and K. Salah, "Iot security: review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.
- [8] M. Andoni, V. Robu, D. Flynn et al., "Blockchain technology in the energy sector: a systematic review of challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 143–174, 2019.
- [9] S. Shi, D. He, L. Li, N. Kumar, M. K. Khan, and K.-K. R. Choo, "Applications of blockchain in ensuring the security and privacy of electronic health record systems: a survey," *Computers & Security*, vol. 97, Article ID 101966, 2020.
- [10] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.
- [11] H. Debiao, C. Jianhua, and Z. Rui, "A more secure authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, no. 3, pp. 1989–1995, 2012.
- [12] Z.-Y. Wu, Y.-C. Lee, F. Lai, H.-C. Lee, and Y. Chung, "A secure authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, no. 3, pp. 1529–1535, 2012.
- [13] S. Kumari, M. K. Khan, and R. Kumar, "Cryptanalysis and improvement of 'a privacy enhanced scheme for telecare medical information systems,'" *Journal of Medical Systems*, vol. 37, no. 4, pp. 1–11, 2013.
- [14] C. L. Chen, T. T. Yang, and T. F. Shih, "A secure medical data exchange protocol based on cloud environment," *Journal of Medical Systems*, vol. 38, no. 9, pp. 112–12, 2014.
- [15] S.-Y. Chiou, Z. Ying, and J. Liu, "Improvement of a privacy authentication scheme based on cloud for medical environment," *Journal of Medical Systems*, vol. 40, no. 4, p. 101, 2016.
- [16] I.-C. Lin, M.-S. Hwang, and L.-H. Li, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.
- [17] W.-S. Wen-Shenq Juang, "Efficient multi-server password authenticated key agreement using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 251–255, 2004.
- [18] J.-L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Computers & Security*, vol. 27, no. 3-4, pp. 115–121, 2008.
- [19] W.-J. Tsaur, J.-H. Li, and W.-B. Lee, "An efficient and secure multi-server authentication scheme with key agreement," *Journal of Systems and Software*, vol. 85, no. 4, pp. 876–882, 2012.
- [20] J.-L. Tsai and N.-W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE systems journal*, vol. 9, no. 3, pp. 805–815, 2015.
- [21] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Systems Journal*, vol. 14, no. 1, pp. 560–571, 2019.
- [22] X. Liu, W. Ma, and H. Cao, "Mbpa: a medibchain-based privacy-preserving mutual authentication in tmis for mobile medical cloud architecture," *IEEE Access*, vol. 7, Article ID 149282, 2019.
- [23] A. Yazdinejad, G. Srivastava, R. M. Parizi, A. Dehghantaha, K.-K. R. Choo, and M. Aledhari, "Decentralized authentication of distributed patients in hospital networks using blockchain," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 8, pp. 2146–2156, 2020.
- [24] C.-T. Li, D.-H. Shih, C.-C. Wang, C.-L. Chen, and C.-C. Lee, "A blockchain based data aggregation and group authentication scheme for electronic medical system," *IEEE Access*, vol. 8, Article ID 173904, 2020.
- [25] X. Cheng, F. Chen, D. Xie, H. Sun, and C. Huang, "Design of a secure medical data sharing scheme based on blockchain," *Journal of Medical Systems*, vol. 44, no. 2, pp. 1–11, 2020.

- [26] C. Lin, D. He, X. Huang, M. Khurram Khan, and K.-K. R. Choo, "A new transitively closed undirected graph authentication scheme for blockchain-based identity management systems," *IEEE Access*, vol. 6, Article ID 28203, 2018.
- [27] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for iiot devices," *IEEE Transactions on Industrial Informatics*, 2021.
- [28] H. Xiong, C. Jin, M. Alazab et al., "On the design of blockchain-based ecdsa with fault-tolerant batch verification protocol for blockchain-enabled iomt," *IEEE Journal of Biomedical and Health Informatics*, 2021.
- [29] L. Harn and H.-Y. Lin, "Integration of user authentication and access control," *IEE Proceedings - Computers and Digital Techniques*, vol. 139, no. 2, pp. 139–143, 1992.
- [30] N.-Y. Nam-Yih Lee, "Integrating access control with user authentication using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 943–948, 2000.
- [31] Y.-C. Chen and L.-Y. Yeh, "An efficient authentication and access control scheme using smart cards," in *11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, vol. 2, pp. 78–82, IEEE, 2005.
- [32] C. Yang, Z. Jiang, and J. Yang, "Novel access control scheme with user authentication using smart cards," in *2010 Third International Joint Conference on Computational Science and Optimization*, vol. 2, pp. 387–389, IEEE, 2010.
- [33] C. Lin, D. He, X. Huang, K.-K. R. Choo, and A. V. Vasilakos, "Bsein: a blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," *Journal of Network and Computer Applications*, vol. 116, pp. 42–52, 2018.
- [34] L. Xiong, F. Li, M. He, Z. Liu, and T. Peng, "An efficient privacy-aware authentication scheme with hierarchical access control for mobile cloud computing services," *IEEE Transactions on Cloud Computing*, 2020.
- [35] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2018.
- [36] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 4957–4968, 2019.
- [37] H. Boyapally, P. Mathew, S. Patranabis et al., "Safe is the new smart: puf-based authentication for load modification-resistant smart meters," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, 2020.
- [38] M. Fakroon, F. Gebali, and M. Mamun, "Multifactor authentication scheme using physically unclonable functions," *Internet of Things*, vol. 13, Article ID 100343, 2021.
- [39] C. C. Chang and J.-Y. Kuo, "An efficient multi-server password authenticated key agreement scheme using smart cards with access control," in *19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers)*, vol. 2, pp. 257–260, IEEE, 2005.
- [40] D. He, S. Zeadally, N. Kumar, and W. Wu, "Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2052–2064, 2016.

Research Article

CTRF: Ethereum-Based Ponzi Contract Identification

Xuezhi He , Tan Yang , and Liping Chen 

State Key Laboratory of Networking and Switching Technology School of Computer Science(National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Tan Yang; tyang@bupt.edu.cn

Received 30 December 2021; Revised 27 January 2022; Accepted 2 March 2022; Published 29 March 2022

Academic Editor: Yuling Chen

Copyright © 2022 Xuezhi He 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, blockchain technology has been developing rapidly. More and more traditional industries are using blockchain as a platform for information storage and financial transactions, mainly because of its new characteristics of non-tamperability and decentralization compared with the traditional systems. As a representative of blockchain 2.0, Ethereum has gained popularity upon its introduction. However, because of the anonymity of blockchain, Ethereum has also attracted the attention of some unscrupulous people. Currently, millions of contracts are deployed on Ethereum, many of which are fraudulent contracts deployed by unscrupulous people for profit, and these contracts are causing huge losses to investors worldwide. Ponzi contracts are typical of these contracts, which mainly reward the funds invested by later investors to early investors, and later investors will have no gain. However, although there are some studies for identifying Ponzi contracts on Ethereum, there is some room for progress in the research. Therefore, we propose a method to detect Ponzi scheme contracts on Ethereum-CTRF. This method forms a dataset by extracting the word features and sequence features of the smart contract's code and the features of transactions. The dataset is divided into a training set and a test set. Oversampling is performed on the training set to deal with the problem of positive and negative sample imbalance. Finally, the model is trained on the training set and tested on the test set. The experimental results show that the model has significantly improved recall compared with existing Ponzi contract detection methods.

1. Introduction

Blockchain technology was proposed by Nakamoto [1], and since its introduction in 2008, it has received widespread attention. Due to its decentralized and tamper-evident features, blockchain technology has now been applied in the fields of finance, healthcare, and social governance. Based on these advantages, blockchain technology will have even broader application prospects in the future.

In the field of financial transactions, the underlying blockchain technology-based electronic cash systems (Bitcoin and Ethereum) have attracted many investors. However, they have also attracted many unscrupulous people. Owing to the inherent anonymity and tamper-evident nature of the blockchain, the Bitcoin and Ethereum platforms have also become a breeding ground for unscrupulous transactions to take place. On the Silk Road website, which was shut down in 2013, as much as \$300,000–\$500,000 per

day was traded in Bitcoin regarding drugs and private data, and at the time of Silk Road's closure in 2013, approximately 9.5 million Bitcoins worth \$1.2 billion had been traded on Silk Road [2].

Ethereum is known as Blockchain 2.0, which mainly solves the problem of lack of scalability of Bitcoin system; however, while people add new features to Ethereum, new risks are also introduced. Phishing, fraud, theft, and other illegal criminal activities launched by taking advantage of the flaws in the Ethereum blockchain technology have emerged. On June 17, 2016, the DAO of Ethereum was attacked by hackers. The attacker exploited a vulnerability in a contract written by the DAO to transfer more than 3 million Ether coins equivalent to \$60 million from its asset pool to its sub-DAO [3].

Among the plethora of scams, Ponzi schemes, a classic scam in the real world, are also happening on the blockchain. Ponzi schemes on Bitcoin usually actively advertise their on-

chain projects on social media as having high rewards and low risk to attract investors. Ponzi schemes on Ethereum are smart contracts written in the high-level programming language, Solidity [4]. They are generally packaged as investment projects or gambling games that also promise huge returns to investors. Some Ponzi schemes even create their promotional websites to attract investors through aggressive marketing campaigns. Usually, investors know little about blockchain, and it can be difficult for them to tell which are meaningful smart contract investment projects and which are smart contracts disguised as high-yield investment schemes. According to Chainalysis, scams on Ethereum from 2017 to 2019 affected millions of people and caused \$4.3 billion worth of losses. The majority of these came from Ponzi schemes, accounting for 92% of the total. The number of victims of Ethereum Ponzi scams alone was 2.4 million, with the average amount transferred by victims being \$1,676 [5]. According to Bartoletti et al., 191 smart Ponzi schemes active on Ethereum raised almost \$500,000 from more than 2,000 different users from August 2015 to May 2017 [6].

It is evident from reading the previously mentioned studies that there is an urgent need to strengthen the regulation and monitoring of the blockchain market. Although there are some studies in this area, they do not focus on the recall value of the experimental results. In fraud detection applications, as in many fields with unbalanced class distributions, it is more important to correctly classify the true classes (i.e., the “Ponzi” classes in our problem) than to correctly classify the majority classes. Therefore, the recall is more important than the precision of the prediction.

We have achieved good results by dealing with the data imbalance problem and by setting up a large number of experiments in different environments with more effective features. The recall values have improved compared to existing studies. To summarize, our contributions are as follows:

- (1) dealing with the imbalance between positive and negative samples of the dataset by expanding positive samples and oversampling,
- (2) evaluation and comparison of models for classifying Ethereum Ponzi schemes, ultimately our model has a higher recall,
- (3) assessment of feature contribution.

2. Related Work

In terms of Ponzi scheme research on Bitcoin, Vasek and Moore analyzed the supply and demand of Bitcoin-based Ponzi schemes, identified 1780 Ponzi schemes, and derived the determinants affecting the life cycle of Bitcoin Ponzi schemes [7]. Boshmaf et al. analyzed MMM, one of the oldest Ponzi schemes on Bitcoin, and proposed analytical criteria and metrics for the Ponzi scheme of cryptocurrencies [8]. It is worth mentioning that they counted the daily Gini coefficients of MMM to measure the income gap between investors. Bartoletti et al. designed a set of relevant characteristics to classify Bitcoin Ponzi schemes, such as average amount invested by users, maximum daily trading

volume, number of active days of contracts, number of users, and Gini coefficients. Metrics such as F-score and AUC were then used to evaluate the effectiveness of different supervised learning classification algorithm models and finally succeeded in finding 31 of the 32 Ponzi schemes [9]. Although the imbalanced dataset was treated by them, the model may still suffer from overfitting by reason of the large gap between positive and negative sample size, so there is still room for improvement in their experiments.

In the study of Ponzi schemes on Ethereum, Zheng et al. surveyed the challenges and recent advances in smart contracts, giving a complete picture of the challenges smart contracts by dividing the smart contract lifecycle into four phases: creation, deployment, execution, and completion, where scams like Ponzi contracts are classified as the last phase of the contract’s lifecycle, and most scams cause harm to contract users during the contract completion phase [10]. Chen et al. analyzed the current problems of Ethereum from three perspectives: vulnerability, attack, and defense. In the paper, the Rubixi contract is used as an example to classify Ponzi contracts as an attack means in the application layer of Ethereum [11]. Hu et al. analyzed the transaction behavior pattern between Ponzi contracts and other scam contracts to classify contracts from the perspective of transactions [12]. Jung et al. used the 0-day model to analyze the model based on the bytecode features of the contract and finally determined whether the contract is a Ponzi contract [13]. However, they did not consider the transaction features of the contract in their experiments, which may lead to a decrease in accuracy compared with the model that incorporates transaction features. Yujian and Bo classified Ponzi contracts into tree-shaped, chain-shaped, waterfall-shaped, and handoff-shaped by analyzing the Ponzi contract source code. They proposed that the similarity between contract bytecodes can be measured by using NLD [14] (Normalized Levenshtein Distance) and setting the corresponding threshold to determine whether two contracts are similar and whether the contract is a Ponzi contract. Subsequently, they measured the impact of Ponzi contract on Ethereum by counting the total transaction amount [6]. Sun et al. were inspired by the flowchart of traditional test domain code; the bytecodes generated during the operation of a Ponzi contract are concatenated and plotted as a tree of invocation behaviors. The model is trained by comparing the similarity of the behavior trees [15]. Fan et al. solved the prediction bias problem which was made of target leakage during training and improved the generalization ability of the model by analyzing the imbalance and repetition of Ponzi contracts in the dataset [16]. Chen et al. extracted the transaction features from the transaction data of smart contracts and combined them with the opcode of smart contracts in extracted opcode frequency features and used XGBoost to train these data features. Eventually, 434 Ponzi contracts were found by detecting contracts deployed before May 7, 2017 [17].

Although the aforementioned studies achieved good results, most of their experiments aimed at improving the model accuracy without considering improving the recall of the model. In contrast, our CTRF (Code and Transaction Random Forest) model improves the recall of the model by

adding sequence features of the opcodes to the code features and extracting more efficient features of transactions.

3. Smart Ponzi Contracts

The definition of smart contract can be traced back to 1994 by Szabo [18]. It was first defined as an alternative to traditional paper contracts and a digital representation of the transaction agreement to help the parties to fulfill the contractual project. However, due to the immaturity of the technology and the lack of a trustworthy platform for execution, smart contracts did not attract much attention. The establishment of blockchain and the rise of decentralized platforms have brought smart contracts back into the public eye. Ethereum is known as blockchain 2.0 differs from blockchain 1.0 in that Ethereum has a Turing-complete programming language [19]. Developers can implement smart contracts in the high-level programming language Solidity or Golang and compile them for deployment on the EVM (Ethereum virtual machine) [4]. Smart contracts are the basis for implementing blockchain-based information systems in various domains. It can execute transactions without a trusted third party by triggering conditions through program code. For example, the following is a simple example of a smart Ponzi contract-0x83Fccc659EeeeE98ca9764B7B34409347DFbc98b from the source code; we can know that every investment received by the contract will transfer 1% of the balance to the contract creator, and this contract every 5900 blocks (24 hours) will pay 5% of the balance of the contract to the investor.

```
pragma solidity ^0.4.24;
contract eternity {
    address pr = 0x587a38954a
D9d4DEd6B53a8F7F28D32D28E6bBD0;
    address ths = this;
    mapping (address => uint) balance;
    mapping (address => uint) paytime;
    mapping (address => uint) prtime;
    function () external payable {
        if ((block.number-prtime[pr]) >= 5900){
            pr.transfer(ths.balance/100);
            prtime [pr] = block.number;
        }
        if (balance[msg.sender] != 0){
            msg.sender.transfer ((block.number-paytime
[msg.sender])/5900*5);balance[msg.sender]/100*5);
        }
        paytime[msg.sender] = block.number;
        balance[msg.sender] += msg.value;
    }
}
```

The smart contract runs on the EVM. After compiling the smart contract code into bytecode and uploading it to Ethereum through transactions, Ethereum will automatically return the generated contract account address, and finally, investors can interact with the contract through transactions [20]. The code of smart contracts can implement a wide variety of functions, which provides investors with a wealth of investment options. However, this can also confuse investors, who may fall into the trap of scams without being fully familiar with these smart contracts.

The Ponzi scheme was invented by Charles Ponzi, an Italian businessman, in which he promised investors a 40% profit return within three months. After attracting investors, he paid the new investors' money as a return to those who initially invested and then enticed more people to invest. He eventually attracted 30,000 investors in seven months. The more official definition from the SEC (United States Securities and Exchange Commission) is "A Ponzi scheme is an investment fraud that involves the payment of purported returns to existing investors from funds contributed by new investors. Ponzi scheme organizers often solicit new investors by promising to invest funds in opportunities claimed to generate high returns with little or no risk. With little or no legitimate earnings, Ponzi schemes require a constant flow of money from new investors to continue. Ponzi schemes inevitably collapse, most often when it becomes difficult to recruit new investors or when a large number of investors ask for their funds to be returned" [21].

From the definition, the typical feature of the Ponzi scheme is to pay the existing investors the so-called returns with the funds provided by the new incoming investors. Compared with other financial frauds, Ponzi schemes are characterized by many victims, wide impact, deep damage, high concealment, and serious social harm. In the Ethereum smart contract, the Ponzi scheme has some new characteristics.

- (1) The most obvious thing is that it is based on the anonymity of the blockchain; people cannot know the real identity of the contract initiator. For the unscrupulous this greatly reduces the risk of them committing a scam, but for the average contract user, the risk of their money being compromised is greatly increased.
- (2) Ponzi contracts are simpler and more efficient than traditional Ponzi schemes. The scammers only need to deploy Ponzi contracts to Ethereum and they can effortlessly reap the benefits when transactions occur.
- (3) Ponzi contracts are easier to replicate and implement. By reviewing the Ponzi contract code, we found that many of the contract codes are identical.
- (4) Due to the anonymity of blockchain and the difficulty of traceability, the defrauded funds cannot be successfully recovered, leaving the investors to suffer losses.

4. Methodology

4.1. Dataset. The dataset used in this paper is based on the open-source shared address set [6, 17, 22]. One of the publicly available address sets in [17] originally contained 3590 common contract addresses and 200 Ponzi contract addresses and three incorrect addresses. In our experiments, we eliminated two non-Ponzi contract addresses that were not successfully deployed and found two addresses with exactly opposite labels to other address sets. After careful inspection of the contract source code, the corrected address set situation is as follows: 3586 common contract addresses and 202 Ponzi contract addresses, a total of 3788 smart contracts, where the ratio of positive to negative cost is 1 : 18. We labeled this address set as D1.

Since the gap between the positive and negative sample ratios in D1 is too large, we first dealt with the positive and negative sample imbalance by expanding the positive sample data. The publicly available address set in [6] contains 184 Ponzi contracts, and after removing two of the duplicate addresses, its correction includes 182 Ponzi contract addresses. We collected another 50 Ponzi contract addresses from [22]. The address sets of [6, 22] are combined and then deduplicated against D1 to finally obtain 96 Ponzi contract addresses, which are added to D1 to obtain the expanded address set D2. The expanded address set D2 contains 3586 ordinary contract addresses and 298 Ponzi contract addresses, totaling 3884 smart contracts, where the ratio of positive to negative samples is 1 : 12.

The address dataset remains unbalanced after expanding the positive samples. Besides expanding the data from the perspective of the data source, the other two solutions to deal with the imbalance of the dataset at present are oversampling and undersampling. Oversampling means balancing the positive and negative sample ratios by generating samples from minority classes. And undersampling means reducing the samples of most classes to balance the positive and negative sample ratios. Here, we chose SMOTE oversampling based on oversampling; the basic idea of SMOTE algorithm is to analyze the minority class samples and synthesize new minority class samples added to the dataset according to the KNN algorithm, thus enriching the number of minority class samples and avoiding the problem of overfitting caused by oversampling by copying minority class samples in the past [23]. As in Figure 1, we constructed the feature dataset in two main ways.

- (1) Get the bytecode of the contract by Etherscan, then disassemble the bytecode into opcode, and finally convert it into code features.
- (2) Obtain the transactions of the contract and calculate the corresponding transaction features, such as life time and Gini coefficient.

The D1-code dataset and D2-code dataset are obtained by extracting opcode features on D1 and D2, respectively. The D1-codeAndTran dataset and D2-codeAndTran dataset

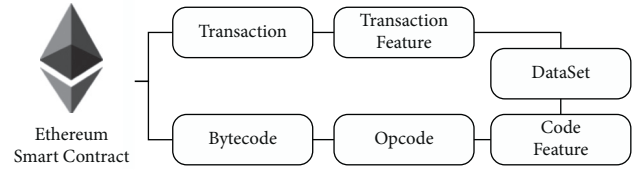


FIGURE 1: Flowchart of feature extraction.

are obtained by merging the opcode features and transaction features. The experiments are mainly focused on these four datasets. The datasets have been open-sourced in this paper, which can be found at <https://github.com/BuptHxz/DetectionOfPonziContract>.

4.2. Feature Extraction

4.2.1. Code Feature. Initially, we wanted to get the code features of the contracts by getting the internal implementation logic and keywords directly from the source code. But since the code of most of the contracts in the dataset is not publicly available, we only got the code features of a very small fraction of the contracts, which made it impossible to build a complete dataset. Although most of the contracts' code is not available, we can get the contracts' bytecode, which is compiled from the code, through Etherscan.io. And according to Kiffer et al. [24, 25], the similarity between contracts can be effectively detected by detecting the similarity of contract bytecode. Therefore, we started from the bytecode and got the code features of the contracts by disassembling and other techniques.

As described in Section 2, if developers want to run a contract on Ethereum, they first need to write a smart contract by Solidity, and then, the code is transformed into bytecode after compiling the code with the corresponding specific version of the compiler. Finally, the compiled bytecode is deployed to Ethereum. The bytecode is represented by a string of hexadecimal codes. The following is the compiled bytecode of the Solidity code shown in Section 3.

“bytecode”：“608060405260008054600160a060020a031990811673587a38954ad9d4ded6b53a8f7f28d32d28e6bbd017909155600180549091163017905534801561004457600080fd5b50610178806100546000396000f30060806040526000805473ffffffffffffffffffffffff1681526004602052604090205461170c4391909103106100b55760005460015473ffffffffffffffffffffffffffffffff918216916108fc916064911631049081150290604051600060405180830381858888f1935050505015801561008b573d6000803e3d6000fd5b506000805473ffffffffffffffffffffffffffffffff1681526004602052604090204390555b336000908152600260205260409020541561012857336000818152600260209081526040808320546003909252909120546108fc9160649161170c43919091030402046005029081150290604051600060405180830381858888f19350505050158015610126573d6000803e3d6000fd5b505b336000908152600360209081526040808320439055600290915290208054340190550000a165627a7a72305820bccf7dff930cd8237a3d56127f741c545a3f33447c34351f3009e937ea335baf0029”

These bytes correspond to EVM operations and thus instruct the EVM to run the code. To make it easy to distinguish them, Ethereum officials convert these bytes into corresponding easy-to-remember opcodes (Opcodes) and record them in the Yellow Book [20]. For example, 0x60 converted to Opcode is PUSH1; 0x80 converted to Opcode is DUP1. The corresponding partial conversions are shown in Table 1.

The bytecode can be converted to an opcode sequence as follows.

“opcodes”: “PUSH1 0x80 PUSH1 0x40 MSTORE PUSH1 0x00 DUP1 SLOAD PUSH1 0x01 PUSH1 0xa0 PUSH1 0x02 EXP SUB NOT SWAP1 DUP2 AND PUSH2 0x587a38954ad9d4ded6b53a8f7f28d32d28e6bbd0 OR SWAP1 SWAP2 SSTORE PUSH1 0x01 DUP1 SLOAD SWAP1 SWAP2 AND ADDRESS OR SWAP1 SSTORE CALLVALUE DUP1 ISZERO PUSH2 0x0044 JUMPI PUSH1 0x00 DUP1 REVERT JUMPDEST POP PUSH2 0x0178 DUP1 PUSH2 0x0054 PUSH1 0x00 CODECOPY PUSH1 0x00 RETURN STOP PUSH1 0x80 PUSH1 0x40 MSTORE PUSH1 0x00 DUP1 SLOAD ...”.

By counting the number of bytecodes corresponding to ordinary contracts and Ponzi contracts, we find that there are significant differences between Ponzi contracts and ordinary contracts in the number of some bytecodes, as shown in Figure 2. It shows the comparison of the average number of opcodes between Ponzi contracts and ordinary contracts. It is clear that Ponzi contracts and ordinary contracts have the same trend in the number of opcodes and overall Ponzi contracts have fewer opcodes than ordinary contracts. By looking at the numerous source codes, we found that Ponzi contracts tend to implement all the functions through less code, while ordinary contracts functions are more abundant and therefore have more code and a higher average number of opcodes.

We obtained the sequence of opcodes corresponding to the contract bytecodes by disassembling the bytecodes into opcodes. Then, we computed the code features of each contract by using the bag-of-words model [26]. There are opcodes such as PUSH1, PUSH2, DUP1, and DUP2, which we combine into one opcode such as PUSH and DUP, and then, do the statistics, and finally, we select a total of 77 code features by combining the calculations.

However, the features extracted by the bag-of-words model do not take into account the sequence of opcodes and ignore the semantic information of opcodes. Therefore, we use the Doc2Vec model to obtain the sequence features and semantic features of the opcodes to make up for the deficiency of the bag-of-words model in extracting the special diagnosis [27]. The final code features are 77 features extracted by the bag-of-words model and 20 features extracted by the Doc2Vec model.

4.2.2. Transaction Features. According to the characteristics of Ponzi schemes, it is known that most of the later investors incurred losses, and only some early investors may get the gains. Past studies tend to extract the features of trading from the perspective of Ethereum, and we added some new

features on top of this. For example, in the Gini coefficient, as the funds of later investors in a Ponzi scheme are often transferred to the accounts of earlier investors, this characteristic of an unbalanced distribution of funds will make the Gini coefficient larger. The Gini coefficient is a number between 0 and 1. The closer the Gini coefficient is to 1, the more unbalanced the distribution of funds is, and the closer it is to 0, the more balanced the distribution of funds is.

We selected the following transaction characteristics:

- (1) Bal: the balance of the contract after the last trade
- (2) TotalGet: the number of all ETH received by the contract
- (3) TotalSend: the number of all ETH sent by the contract
- (4) MaxSend: the maximum amount of ETH sent in a single contract
- (5) AvgFee: the average cost of all transactions in the contract
- (6) LifeTime: the survival time of the contract
- (7) GetDivSend: TotalGet/TotalSend
- (8) AddrGetProfit: the number of addresses that receive proceeds from the addresses traded with the contract
- (9) Gini: Gini coefficient

4.3. CTRF Model. We proposed CTRF (Code and Transaction Random Forest) to identify and classify our contracts. The specific CTRF model structure diagram is shown in Figure 3.

In the data preprocess phase, we first obtained the contract transactions and bytecode and then disassembled the bytecode to obtain the contract opcodes. Inspired by NLP, we used the bag-of-words algorithm to obtain the word features of the opcodes and then used the Doc2Vec algorithm to obtain the sequence features of the opcodes. The word features of the opcode and the sequence features together form the code features of the contract. For transaction features, we chose such as Gini coefficient to represent the contract.

In the model train phase, we first synthesized a new sample of Ponzi contracts by SMOTE oversampling, thus solving the positive and negative sample imbalance problem. Subsequently, we composed decision trees by randomly selecting the code features and transaction features of the contract and used the idea of integrated learning to statistically vote on the results using a forest composed of decision trees to obtain the classification results. Finally, we got our training model.

In the model test phase, we inputted the test set into our model trained in the previous phase to finally get the classification results of the contract.

In order to improve the recall value of the model as much as possible, we did the following. First, we solved the problem of imbalance in the number of positive and negative samples in the dataset by expanding the samples of Ponzi contracts. Then, we extracted the word features and

TABLE 1: Examples of converting bytecode to opcode.

Bytecode	Opcode	Bytecode	Opcode
0x60	PUSH1	0x5b	JUMPDEST
0x80	DUP1	0x52	MSTORE
0x56	JUMP	0x50	POP
0x90	SWAP1	0x17	OR

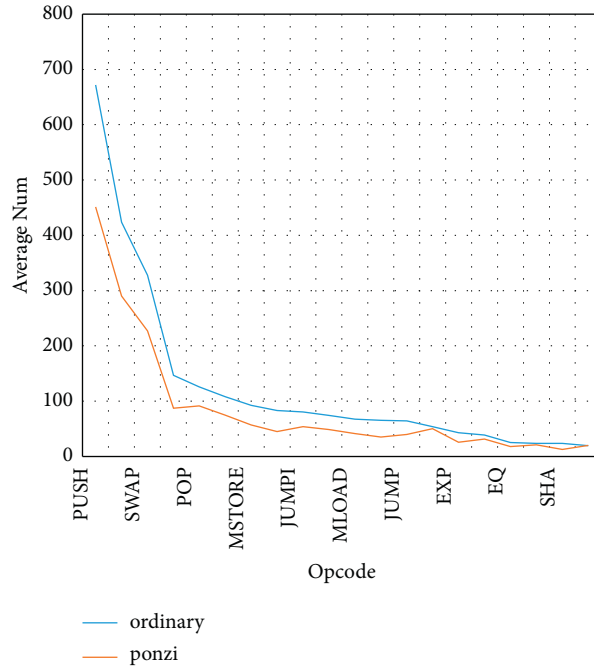


FIGURE 2: The average number of opcodes for ordinary contracts vs. Ponzi contracts.

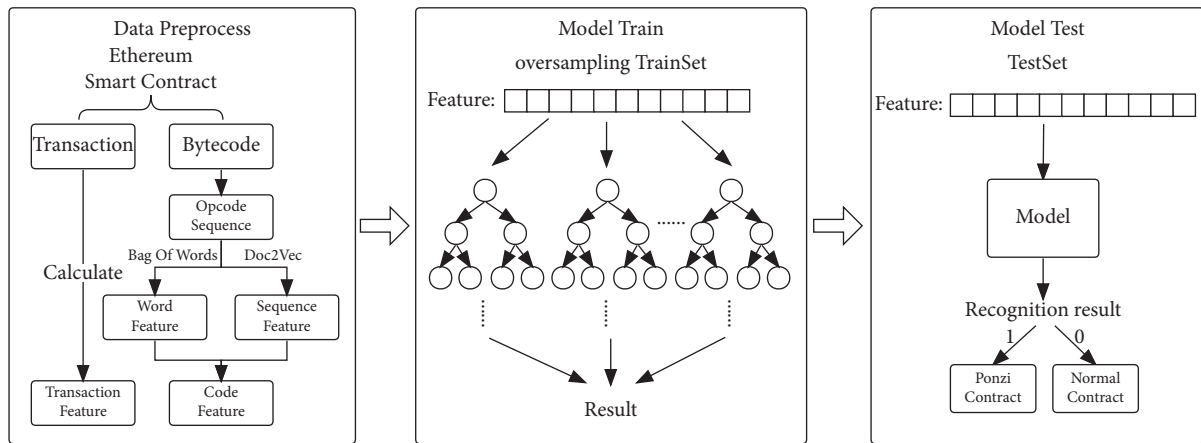


FIGURE 3: CTRF model.

sequence features of the code inspired by NLP in the feature extraction stage, and designed more effective transaction features to represent the Ponzi contracts better. Next, we performed SMOTE oversampling on the positive samples of the dataset to further increase the samples of a few classes and to avoid the overfitting problem of the model. We then penalized its classification errors more severely by increasing

the weights of the Ponzi contract classes in the training phase of the model.

The dataset for this experiment contains a total of 3884 data, which is not enough for deep learning. We tested some deep learning algorithms on the D2 dataset, and the result is that the overfitting of the model leads to poor generalization ability. For this paper, we need to

divide the samples into Ponzi contracts and ordinary contracts, which is equivalent to a dichotomous problem. Therefore, we chose KNN, CNN, DT, SVM, and XGBoost to compare with CTRF and used these six models to experiment on the D1-code dataset, D2-code dataset, D1-codeAndTran dataset, and D2-codeAndTran dataset, respectively. KNN classification algorithm is one of the simplest methods in data mining classification techniques. The so-called K nearest neighbors, which means K nearest neighbors, means that each sample can be represented by its closest K neighbor values to achieve classification. The CNN algorithm is widely used in the field of graph classification, but in recent years it has also been applied to the field of NLP. So, we applied CNN to detect Ponzi contract for comparing with CTRF. DT (decision tree) is a tree-structured algorithm that starts from the root node according to the corresponding features and thus selects branches until it reaches the leaf nodes, taking the category stored in the leaf nodes as the decision result. SVM (support vector machine) is a class of generalized linear classifier that performs binary classification of data in a supervised learning manner, where the decision boundary is the maximum margin hyperplane solved for the learned samples, and the elements are classified after this plane is finally determined. XGBoost is one of the Boosting algorithms. The idea of boosting algorithm is to integrate many weak classifiers to form a strong classifier. Since XGBoost is a boosting tree model, it is integrating many tree models to form a very strong classifier. And the tree model used is the CART regression tree model.

5. Experimental Results and Feature Analysis

5.1. Experiment Setting. Datasets. To compare the validity of the datasets and features, we did experiments on four main datasets.

- (1) D1-code: code features extracted from the corrected Chen's address set as the dataset
- (2) D1-codeAndTran: a dataset consisting of code features and transaction features extracted from the modified Chen's dataset
- (3) D2-code: code features extracted from the expanded dataset as a dataset
- (4) D2-codeAndTran: a dataset consisting of code and transaction features extracted from the expanded dataset

We conducted independent experiments on these four datasets: first cross-validating to find the best experimental parameters, then using 70% of the dataset for training and 30% for testing, and finally conducting 20 experiments to calculate the average results.

Evaluation Metrics. In this paper, precision, recall, and F-score are used as the evaluation criteria for the experimental results. Among them, precision is the proportion of all contracts judged as a certain category that are contracts of that category. The recall is the proportion

of the number of detected Ponzi contracts to the total number of Ponzi contracts. The F-score is a summation of the precision and recall values. The solution formula for the three selected metrics is shown in (1)–(3).

$$\text{Precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}, \quad (1)$$

$$\text{Recall} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}, \quad (2)$$

$$F\text{-score} = 2 \times \frac{\text{Precision}}{\text{Precision} + \text{Recall}} \quad (3)$$

True positive is the number of Ponzi contracts that are correctly determined. False positive is the number of non-Ponzi contracts that are misclassified as Ponzi contracts. False negative is the number of Ponzi contracts that are misclassified as non-Ponzi contracts.

5.2. Results Summary. Table 2 summarizes the results of the corresponding features of the original and expanded datasets under different methods. After analyzing the data in the table, we got the following conclusions.

It is clear that CTRF outperforms other algorithms in terms of precision and recall, and CTRF and XGBoost also outperform the original dataset D1 on the D2 dataset after our positive sample expansion. Although KNN, CNN, DT, and SVM perform well in terms of recall, their precision is poor.

D1-code. Since we added the sequence feature of the opcode, the experimental results improve the recall value by 11% and the F1 value by 7% compared with those in [28]. This indicates that sequence features of the opcode can help the model to identify more Ponzi contracts.

D1-codeAndTran. The recall value obtained in the experiment of Chen et al. was 0.69 [28]. In contrast, we achieve a recall of 0.85, which is a full 16% improvement. Our selected transaction features are better than those selected by Chen et al., and the recall of the experiment improves slightly after the inclusion of the transaction features. The addition of transaction features does enhance the model's identification of Ponzi contracts.

D2-code. After expanding the data, the recall value is improved by 3%. This means the imbalance between positive and negative samples affects the effect of the model. Therefore, CTRF shows a higher recall value in the D2 dataset after we expand the positive sample and oversample it.

D2-codeAndTran. After adding transaction features to the extended dataset, the experimental results are slightly improved compared with those on the extended dataset D2-code without transaction features, where the recall value is improved by 2%. It proves that our extracted transaction features such as Gini coefficients can indeed help the model identify more Ponzi contracts.

TABLE 2: The experimental results on four datasets.

Metric		KNN	CNN	DT	SVM	XGBoost	CTRF
Precision	D1-code	0.518	0.486	0.500	0.586	0.918	0.953
	D2-code	0.501	0.630	0.551	0.543	0.926	0.933
	D1-codeAndTran	0.485	0.621	0.571	0.642	0.907	0.929
	D2-codeAndTran	0.478	0.705	0.611	0.545	0.918	0.928
Recall	D1-code	0.803	0.583	0.836	0.836	0.811	0.847
	D2-code	0.813	0.773	0.812	0.788	0.863	0.875
	D1-codeAndTran	0.787	0.683	0.721	0.852	0.828	0.852
	D2-codeAndTran	0.801	0.761	0.863	0.763	0.873	0.891
F-score	D1-code	0.628	0.530	0.626	0.689	0.862	0.897
	D2-code	0.619	0.694	0.657	0.643	0.893	0.903
	D1-codeAndTran	0.601	0.651	0.638	0.732	0.865	0.889
	D2-codeAndTran	0.598	0.732	0.715	0.635	0.894	0.909

TABLE 3: The importance of the twenty most significant features.

	D1-code	D2-code	D1-codeAndTran	D2-codeAndTran
1	LT	LT	LT	LT
2	SLOAD	LOG	LOG	maxSend
3	LOG	CALLDATALOAD	AND	totalSend
4	GAS	SLOAD	maxSend	addrGetPro
5	AND	CALL	SLOAD	avgFee
6	CALLDATALOAD	AND	MSTORE	LOG
7	SSTORE	RETURN	totalSend	Gini
8	CALL	STOP	CALLDATALOAD	AND
9	MSTORE	SUB	MUL	SLOAD
10	MUL	GAS	SSTORE	CALL
11	GT	MSTORE	SHA	CALLDATALOAD
12	SHA	MUL	GAS	RETURN
13	DUP	RETURN	SUB	totalGet
14	RETURN	CALLDATASIZE	DUP	GAS
15	SUB	SSTORE	GT	STOP
16	STOP	GT	STOP	SUB
17	TIMESTAMP	CALLVALUE	avgFee	MSTORE
18	ADD	CODECOPY	CODECOPY	SHA
19	MLOAD	MLOAD	CALL	GT
20	OR	EXTCODESIZE	ISZERO	MUL

5.3. *Feature Analysis.* We obtained the corresponding feature importance rankings by analyzing the performance of the CTRF model on the four datasets, as shown in Table 3.

On D1-code and D2-code, our operand sequence features (D2V) extracted by Doc2Vec can better help the model to identify the Ponzi contracts. Among the more important word features extracted by bag-of-words are LT and log.

LT represents less than judgment in EVM. After calculation, we concluded that on average each Ponzi contract has 14 LT opcodes, while each non-Ponzi contract has only 11 on average. And most of the Ponzi contracts have fewer opcodes corresponding to them than non-Ponzi contracts. Only the number of LT opcode is more in Ponzi contracts than in non-Ponzi contracts, which is why LT opcode is so important for detecting Ponzi contracts. The importance of LT opcode is also reflected in the Ponzi contract code. We observed a large number of Ponzi contracts and found that when most of them receive a transfer from an external

account, they will determine whether the amount of the transfer is less than the minimum investment threshold set by the contract. If it is less than that, the investment will be swallowed directly and no subsequent returns will be made to the investor.

During the training process of the D2-codeAndTran dataset, the importance of our newly added transaction features is located in the top positions, and according to the test results, the newly added transaction features make the model outperform the D2 code, which ultimately leads to a 2% improvement in the recall value. It is worth mentioning that in [28] after Chen et al. added transaction features to the dataset, the accuracy of the model improved slightly, but the recall of the model decreased by 4%. In our experiments, on the other hand, we have significantly improved the recall of the model by adding transaction features such as Gini coefficients. Compared with the transaction features extracted by Chen et al., our extracted transaction features are more helpful for the model to identify Ponzi contracts.

6. Conclusion

Nowadays, the issue of on-chain security on Ethereum is attracting more and more attention. Some researches on Ethereum security have also emerged. In this study, we extracted the classification model CTRF for the identification and analyzed of Ponzi schemes on Ethereum. First, we relied on increasing the number of positive samples to get the original dataset D1 and the expanded dataset D2 and then extracted the code features and transaction features of the two datasets, respectively, to get four datasets D1-code, D2-code, D1-codeAndTran, and D2-codeAndTran, and each dataset is divided into training and testing sets according to the ratio of 7:3. The training set of the four datasets is then oversampled to deal with the problem of positive and negative sample imbalance. Finally, the corresponding models are trained on each of the four training sets and tested on the test set to obtain the test results.

From the test results, the expanded dataset D1-codeAndTran, the recall value is improved by about 16% compared with the results in [28]. And the model is still able to produce good results without transaction features, and adding our extracted transaction features improves the recall value of the model identification.

In the future, we will make a deeper study on the identification of Ethereum Ponzi contracts. We expect to extract serialized features from the bytecodes of contracts from a deep learning perspective, and then, train them. Then, by comparing the similarity of bytecode sequences between contracts, we can identify Ponzi contracts. In addition, we also expect to build an Ethereum Ponzi contract detection platform to identify and record Ponzi contracts on Ethereum, so as to prevent investors from being cheated. In conclusion, in the future, we will continue our research on Ponzi contracts on Ethereum and maintain the safe and stable development of the Ethereum system.

Data Availability

The datasets mentioned in the article are available at <https://github.com/BuptHxz/DetectionOfPonziContract>.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under grant no. 2019YFC1521101.

References

- [1] S. Nakamoto and A. Bitcoin, "Bitcoin: a peer-to-peer electronic cash system," *Bitcoin*, 2008, <https://bitcoin.org/bitcoin.pdf>.
- [2] J. T. Lawrence, "Virtual currencies; bitcoin & what now after liberty reserve, silk road, and mt. gox?" *Richmond Journal of Law and Technology*, vol. 20, no. 4, 2014.
- [3] E. Gün Sirer, "Thoughts on the dao hack," 2016, <http://hackingdistributed.com/2016/06/17/thoughts-on-the-dao-hack/>.
- [4] "Solidity official documentation," 2020, <https://docs.soliditylang.org/en/v0.8.3/>.
- [5] "Chainalysis," 2020, <https://go.chainalysis.com/2020-Crypto-Crime-Report.html> Crypto crime report.
- [6] M. Bartoletti, S. Carta, T. Cimoli, and R. Saia, "Dissecting ponzi schemes on ethereum: identification, analysis, and impact," *Future Generation Computer Systems*, vol. 102, pp. 259–277, 2020.
- [7] M. Vasek and T. Moore, "Analyzing the bitcoin ponzi scheme ecosystem," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 101–112, Springer, Nieuwpoort, Curaçao, February 2018.
- [8] Y. Boshmaf, C. Elvitigala, H. Al Jawaheri, P. Wijesekera, and M. Al Sabah, "Investigating mmm ponzi scheme on bitcoin," in *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pp. 519–530, Taipei Taiwan, October 2020.
- [9] M. Bartoletti, B. Pes, and S. Serusi, "Data mining for detecting bitcoin ponzi schemes," in *Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pp. 75–84, IEEE, Zug, Switzerland, June 2018.
- [10] Z. Zheng, S. Xie, H.-N. Dai et al., "An overview on smart contracts: challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020.
- [11] H. Chen, M. Pendleton, N. Laurent, and S. Xu, "A survey on ethereum systems security: vulnerabilities, attacks, and defenses," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–43, 2020.
- [12] T. Hu, X. Liu, T. Chen et al., "Transaction-based classification and detection approach for ethereum smart contract," *Information Processing & Management*, vol. 58, no. 2, Article ID 102462, 2021.
- [13] E. Jung, M. Le Tilly, A. Gehani, and Y. Ge, "Data mining-based ethereum fraud detection," in *Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 266–273, IEEE, Atlanta, GA, USA, July 2019.
- [14] L. Yujian and L. Bo, "A normalized levenshtein distance metric," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.
- [15] W. Sun, G. Xu, Z. Yang, and Z. Chen, "Early detection of smart ponzi scheme contracts based on behavior forest similarity," in *Proceedings of the 2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*, pp. 297–309, IEEE, Macau, China, December 2020.
- [16] S. Fan, S. Fu, H. Xu, and C. Zhu, "Expose your mask: smart ponzi schemes detection on blockchain," in *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, IEEE, Glasgow, UK, July 2020.
- [17] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting ponzi schemes on ethereum: towards healthier blockchain technology," in *Proceedings of the 2018 world wide web conference*, pp. 1409–1418, Lyon France, April 2018.
- [18] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.
- [19] V. Buterin, "Ethereum white paper," *GitHub repository*, vol. 1, pp. 22–23, 2013.
- [20] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.
- [21] "Authoritative definition of ponzi scheme," <https://www.sec.gov/spotlight/enf-actions-ponzi.shtml>.

- [22] E. Etherscan, "Ponzi contract," <https://etherscan.io/accounts/label/ponzi>.
- [23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [24] L. Kiffer, D. Levin, and A. Mislove, "Analyzing ethereum's contract topology," in *Proceedings of the Internet Measurement Conference 2018*, pp. 494–499, Boston MA USA, October 2018.
- [25] L. Han, Z. Yang, Yu Jiang, W. Zhao, and J. Sun, "Enabling clone detection for ethereum via smart contract birthmarks," in *Proceedings of the 2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC)*, pp. 105–115, IEEE, Montreal, QC, Canada, May 2019.
- [26] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 43–52, 2010.
- [27] J. HanLau and T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," 2016, <https://arxiv.org/abs/1607.05368>.
- [28] W. Chen, Z. Zheng, E. C.-H. Ngai, P. Zheng, and Y. Zhou, "Exploiting blockchain data to detect smart ponzi schemes on ethereum," *IEEE Access*, vol. 7, pp. 37575–37586, 2019.

Research Article

PPSEB: A Postquantum Public-Key Searchable Encryption Scheme on Blockchain for E-Healthcare Scenarios

Gang Xu ^{1,2}, Shiyuan Xu ¹, Yibo Cao,¹ Fan Yun,¹ Yu Cui,¹ Yiying Yu,¹ and Ke Xiao ¹

¹School of Information Science and Technology, North China University of Technology, Beijing 100144, China

²Advanced Cryptography and System Security Key Laboratory of Sichuan Province, Chengdu 610025, China

Correspondence should be addressed to Ke Xiao; xiaoke@ncut.edu.cn

Received 17 November 2021; Revised 11 December 2021; Accepted 9 March 2022; Published 29 March 2022

Academic Editor: A. Peinado

Copyright © 2022 Gang 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 current E-healthcare scenarios, medical institutions are used to encrypt the information and store it in an Electronic Health Record (EHR) system in order to ensure the privacy of medical information. To realize data sharing, a Public-key Encryption with Keyword Search (PEKS) scheme is indispensable, ensuring doctors search for medical information in the state of ciphertext. However, the traditional PEKS scheme cannot resist the keyword guessing quantum computing attacks, and its security depends on the confidentiality of the secret key. In addition, classical PEKS hand over the search process to a third party, affecting the search results' accuracy. Therefore, we proposed a postquantum Public-key Searchable Encryption scheme on Blockchain (PPSEB) for E-healthcare scenarios. Firstly, we utilized a lattice-based cryptographic primitive to ensure the security of the search process and achieve forward security to avoid key leakage of medical information. Secondly, we introduced blockchain technology to solve the problem of third-party untrustworthiness in the search process. Finally, through security analysis, we prove the correctness and forward security of the solution in the E-healthcare scenarios, and the comprehensive performance evaluation demonstrates the efficiency of our scheme compared with other existing schemes.

1. Introduction

In the current medical scenarios, medical institutions generate a large amount of patient medical data. These data are difficult to supervise, lack necessary technical support, and cost medical institutions many resources. To solve this problem, many medical institutions have adopted EHR systems to reduce the burden and cost of maintaining medical information [1]. The EHR system is a digital health file with medical information as the main body and information sharing as the core. It aims to realize that patients can manage their medical data, and doctors can also access the patient's medical data if they have permission. However, outsourcing management of the EHR system is not an ideal choice. Because the third-party organization responsible for storing the EHR system has too much power, once a malicious attacker buys it, it can launch a collision attack on the medical data in the system to threaten the privacy of medical data. To avoid this situation, medical institutions

usually encrypt medical data through various encryption schemes [2] and store it in the EHR system. Therefore, how to realize the sharing of medical data between patients and doctors in the ciphertext state is a problem to be solved. Thus, Public Key Encryption with Keyword Search (PEKS) [3] is a marvelous candidate in cloud-assisted E-healthcare scenarios, realizing medical data retrieval without privacy leakage. As efficient encryption primitive, it ensures searchable encrypted medical data through keywords.

Although the existing proposed PEKS schemes [4–6] have brought significant benefits to the Internet of Things, there are four significant obstacles to the widespread PEKS in systems in recent decades. Initially, most PEKS schemes were established based on traditional hardness cryptography problems. Nevertheless, with the advent of quantum computers [7] and quantum information [8], the PEKS scheme will be threatened exponentially. Recent breakthrough articles [7] indicate that shortly, it is possible to adopt quantum computers in a realistic view, putting forward higher

requirements for postquantum cryptographic searchable encryption schemes than before. Secondly, the most computational cost of cloud servers is to search target data from the third-party service agency since cloud servers need to execute a verification procedure for the corresponding keyword. Due to the exorbitant public-key encryption operations, the existing PEKS scheme introduces a significant calculation overhead. In the E-healthcare scenarios, the cloud server can work with medical data from mobile medical detection devices simultaneously to retrieve the data of multiple doctors. Therefore, it has a performance bottleneck on the medical cloud servers. Thirdly, with the explosive utilization of mobile medical detection equipment, most schemes have key exposure problems [9]. The existing PEKS scheme cannot guarantee the forward privacy of the key. The existing PEKS scheme cannot guarantee the forward privacy of the key. Once the doctor's secret key is compromised, the attacker can trace the trapdoor content previously submitted by the doctor, thereby further infringing on the confidentiality of the outsourced data [10]. In this regard, we optimize the lattice cryptography in our scheme to make the key have relations with period to ensure that the key exposure at the previous period will not affect the medical data confidentiality at the later period and achieve the forward security of the key [11]. Last but not least, the search function of the traditional PEKS scheme is generally delivered to the service party. However, the untrustworthiness of the service party will cause attackers to generate Keyword Guess Attacks (KGA) on medical information. Fortunately, blockchain can effectively solve this problem [12–17]. Blockchain is a new database technology that can realize decentralized distributed architecture design. Its core technical concept was proposed by Satoshi Nakamoto [18] in 2008. Blockchain, as a distributed public ledger, records all transactions packaged in the block without the need for third-party control and ensures the safety and traceability of each transaction record [19]. After a single block is generated, all nodes in the blockchain network use a consensus algorithm to determine whether the block is on the chain, and each block is connected by a hash function, thereby effectively ensuring the immutability of transaction information. Therefore, using blockchain technology to replace the service party in PEKS is an effective way to solve the problem of the untrustworthiness of the service party. For example, [20] replaces the traditional centralized server with a decentralized blockchain system, supports forward and backward privacy, and realizes privacy protection. [21] proposed a novel PEKS scheme, which eliminates the reliance on third-party institutions and makes the entire program completely decentralized. Therefore, to solve the above-mentioned hindrances, we propose a postquantum public-key searchable encryption on blockchain for cloud-assisted E-healthcare scenarios, called PPSEB, based on lattice cryptography [22, 23], one of the postquantum cryptographic primitives, ensuring a robust security level. In addition, we reduce the security of PPSEB to the Learning WithError (LWE) hardness

assumption, which can oppose keyword guessing attacks based on quantum computing launched by malicious attackers effectively.

In our proposed scheme, the patient initially encrypts medical data and its keywords under the public key of the doctor and transmits the corresponding ciphertext to the cloud server for storage. Then, the medical doctor will utilize his/her secret key to compute a trapdoor corresponding to the keyword and then uploads it to the blockchain. Further, the smart contracts on blockchain search for the keyword ciphertext corresponding to the trapdoor and return its number to the cloud server. Finally, the cloud server sends the ciphertext of medical information matching the keyword to the doctor. In summary, we elaborate our main contributions as follows:

- (1) We propose a postquantum Public-key Searchable Encryption on Blockchain (PPSEB) for the E-healthcare scenarios. PPSEB is constructed on lattice-based public-key searchable encryption based on the LWE hardness assumption.
- (2) We then introduce blockchain technology into our proposed scheme in response to the untrustworthiness of third parties during the search process. Therefore, we achieve the decentralization architecture of the PPSEB oracle and enhance the security level.
- (3) PPSEB achieves forward security in order to solve the key leakage of various existing public-key searchable encryption algorithms.
- (4) We give the computational proof of the correctness and forward security of PPSEB. Furthermore, the comprehensive implementation performance evaluation represents that our scheme is efficient in terms of testing time and computational cost compared with existing outperforming E-healthcare schemes and is suitable for medical scenarios.

The structure of our paper is organized as follows. In Section 2, we propose the design goals and security models of our scheme, considering three existing challenges for the proposed PPSEB scheme and the solution to make PPSEB work better in the medical scenarios. In Section 3, we propose our preliminaries of lattice and trapdoor. In Section 4, we present our PPSEB scheme and the main steps of our scheme, including, *PPSEB.Initialization*, *PPSEB.KeyExt*, *PPSEB.Encrypt*, *PPSEB.PEKS*, *PPSEB.Trapdoor*, *PPSEB.Verification*, and *PPSEB.Decrypt*. In Section 5, we provide the security analysis of PPSEB based on correctness and provable security. In Section 6, a precise performance evaluation is proposed by our paper. Finally, we conclude this paper in Section 7.

2. Design Goals and Security Models

2.1. Design Goals. In this paper, we propose three existing challenges for the proposed PPSEB scheme:

- (1) How to make PPSEB resistant to the untrustworthy problem of the service party. In the traditional

searchable encryption scheme, a third-party organization is generally responsible for searching medical information, which makes malicious attackers collude with third-party organizations to provide unreliable search results. Therefore, we use blockchain to replace traditional third-party agencies.

- (2) How to achieve the forward security of PPSEB. Key exposure is a thorny problem faced by existing searchable encryption schemes. Once the private key of the doctor is lost, the attacker can forge the doctor to initiate an inquiry for medical information, and the privacy of medical information cannot be guaranteed. Therefore, how to use lattice-based cryptography to ensure that the leakage of the master key used at this time will not result in the leakage of the past session key is a problem to be solved.
- (3) How to realize PPSEB to resist KGA under quantum computing. The existing searchable encryption scheme cannot guarantee the security of the search process under the attack of quantum computing, and there is a significant commonality between the keywords of medical information. Once the attacker is equipped with a quantum computer, it is possible to launch KGA on medical information through quantum computing, which severely threatens the blockchain system based on traditional cryptography and then exposes the private information contained in the medical information. Consequently, resisting KGA launched by quantum opponents is also a challenging problem. In order to make PPSEB work better in the medical scenarios, the solution in this article should have the following characteristics:
 - (1) Postquantum KGA: PPSEB can resist KGA attacks under quantum computing.
 - (2) Forward security: PPSEB achieves forward security to solve the problem of private key exposure.
 - (3) Efficiency: PPSEB has a higher computational efficiency by reducing the size of the trapdoor.

2.2. Security Model. In this section, we show the ciphertext indistinguishability of our scheme. We can describe several scenarios through games between challenger S and adversary A, in which S generates system security public parameters, initializes the public keys of patient and doctor. A will receive them from S and is permitted to access the oracles as below.

Hash Oracle(HO): A has been permitted to access all values of HO in time t , where $t = 1, 2, \dots, \eta$ and is the total number in the period. Then, A will receive the corresponding hash value.

Break-in phase: After obtaining the query about $SK_{r||t}$ of the doctor in time t by A, S will return the corresponding $SK_{r||t}$ in t time to A. We note that t^* is the break-in period, which satisfies $t > t^*$.

Trapdoor Oracle(TO): A inputs a keyword w to ask S for a trapdoor T_w . Then, we make the restriction $t > t^*$ in order to make sure the forward security, where t^* is break-in period.

Challenge phase: A takes (w_0^*, w_1^*) in t^* and then submits them to S to be the challenge keywords. S then selects b at random and obtains $CT_{t^*}^*$. Consequently, S returns $CT_{t^*}^*$ to A.

Guess phase: At last, A will output $b' \in \{0, 1\}$. It wins the game iff $b' = b$. We define A $d\nu_A^S(k) = |\text{Prob}[b' = b] - 1/2|$, which means the benefit of A to distinguish ciphertexts in t^* successfully.

3. Preliminary

Definition 1 (Lattice). Let $A = [a_1, a_2, \dots, a_n] \in \mathbb{R}^m$ be n linearly independent vectors in m -dimensional space. A lattice L is composed of the linear combination of all integer coefficients of a_1, a_2, \dots, a_n , and we can define: $L(A) = \{\sum_{i=1}^n x_i a_i : i = 1, 2, \dots, n, x_i \in \mathbb{Z}\}$, a_1, a_2, \dots, a_n is known as a basis of L . Given a prime number q , a matrix $A \in \mathbb{Z}_q^{n \times m}$, we define $L_q(A) = \{y \in \mathbb{Z}^m : y = A^T x \text{ mod } q, x \in \mathbb{Z}\}$, $L_q^\perp(A) = \{y \in \mathbb{Z}^m : Ay = 0 \text{ mod } q\}$.

Definition 2 (LWE). Assume q be a prime number, given a random matrix $A \in \mathbb{Z}_q^{n \times m}$, vector $b \in \mathbb{Z}_q^m$ and the error distribution D on \mathbb{Z}_q , find that the vector $s \in \mathbb{Z}_q^n$ satisfies $b = A^T s + e \text{ mod } q$, where $e \in D^m$.

Definition 3 (Statistical Distance). Given two variables X, Y over a domain D , we define the statistical distance of X and Y : $D(X, Y) = 1/2 \cdot \sum_{b \in D} |\text{Pr}[X = a] - \text{Pr}[Y = a]|$.

Definition 4 (Discrete Gaussian Distribution). Let $\rho_{c,\sigma}(x) = \exp -\pi \|x - c\|^2 / \sigma^2$ be the standard

The Gaussian function c represents the center and σ represents the standard deviation. Then we define: $D_{L,c,\sigma}(x) = \rho_{c,\sigma}(x) / \rho_{c,\sigma}(L)$, which is a Gaussian Distribution over Lattice L .

Lemma 1 (TrapGen) [24]. Let $q \geq 3$, $m \geq 2n \log q$. There is a polynomial-time algorithm TrapGen, which outputs a matrix $A \in \mathbb{Z}_q^{n \times m}$ statistically close to the uniform distribution and a trapdoor base $Tr_A \in \mathbb{Z}_q^{m \times m}$, such that $\|Tr_A\| \leq O(n \log q)$ and $\|\widehat{Tr}_A\| \leq O(\sqrt{n \log q})$.

Lemma 2 (SamplePre) [25]. Given $L_q^\perp(A)$, a trapdoor base $Tr_A \in \mathbb{Z}_q^{m \times m}$, a parameter $s \geq \|Tr_A\| \omega(\sqrt{\log m})$, and a vector $v \in \mathbb{Z}_q^n$. Then, the SamplePre algorithm outputs a vector w statistically close to $D_{L_q^\perp(A),s}$, such that $Aw = v \text{ mod } q$.

Lemma 3 (SampleL) [26]. Set a positive integer $m > n$, $q \geq 3$. Given $L_q^\perp(A)$ and its trapdoor base T_A , matrix $B \in \mathbb{Z}_q^{n \times m'}$, parameter $s \geq \|\widehat{T}_A\| \omega(\sqrt{\log(m+m')})$, and vector $u \in \mathbb{Z}_q^n$. The SampleL algorithm computes $e \in \mathbb{Z}_q^{m+m'}$ statistically close to $D_{L_q^\perp(A|B),s}$ such that $(A|B)e = u \text{ mod } q$.

Lemma 4 (SampleR) [26]. Set a positive integer $m > n$, $q \geq 3$. Given $L_q^\perp(B)$ and its trapdoor base $T_B \in \mathbb{Z}^{m \times m}$, matrix $A \in \mathbb{Z}^{n \times m}$, $R \in \mathbb{Z}_q^{n' \times m}$, $s \geq \|\widehat{T}_B\|s' \omega(\sqrt{\log m})$ and vector $u \in \mathbb{Z}_q^n$. The SampleR algorithm outputs a vector $e \in \mathbb{Z}^{m+m'}$ over $D_{L_q^\perp(A|AR+B),s}$ and satisfies $(A|AR+B)e = u \pmod q$, where $s' = \max_{\|x\|=1} \|Rx\|$.

Lemma 5 (NewBasisDel) [27]. Set a positive integer $m > 2n \log q$, $q \geq 3$. Given $L_q^\perp(A)$ and a trapdoor base $T_A \in \mathbb{Z}^{m \times m}$, an invertible matrix $R \in D_{m \times m}$, $D_{m \times m}$ is invertible on $\mathbb{Z}_q^{m \times m}$, $s \geq \|\widehat{T}_A\| \cdot \sqrt{n \log q} \cdot \omega(\sqrt{\log m}) \cdot \sqrt{m} \cdot \omega(\log^{1.5} m)$. The NewBasisDel algorithm outputs $L_q^\perp(B)$ and a trapdoor base $T_B \in \mathbb{Z}^{m \times m}$ responding to $L_q^\perp(B)$, where $B = AR^{-1}$.

Lemma 6 (SampleRwithBasis) [27]. Given a positive integer $m > 2n \log q$, $q \geq 3$, and a random matrix $A \in \mathbb{Z}_q^{n \times m}$, its column vector can generate \mathbb{Z}_q^n . The Sample R with Basis algorithm outputs an invertible matrix $R \in D_{m \times m}$, a lattice $L_q^\perp(B)$ and its trapdoor base $T_B \in \mathbb{Z}^{m \times m}$, where $B = AR^{-1} \pmod q$, T_B satisfies $\|\widehat{T}_B\| \leq O(\sqrt{n \log q})$.

Definition 5. (PEKS scheme): One general PEKS scheme includes five algorithms as PEKS = (Initialization, KeyExt, PEKS, Trapdoor, Verification), these algorithms are defined in the following sentences:

$(s, X) \leftarrow \text{Initialization}(\perp)$: In this step, it generally initializes some security parameters s , and parameters regard to the Gaussian Distribution X in one time period j . The output is just these parameters which will utilize in the next step.

$(pk, sk) \leftarrow \text{KeyExt}(s)$: After inputting the parameter s , it will output the public key pk and secret key sk , which consist (pk, sk) .

$s_\varepsilon \leftarrow \text{PEKS}(pk, \varepsilon)$: The algorithm takes a public key pk and one keyword ε as input, and outputs a ciphertext s_ε of ε .

$t_\varepsilon \leftarrow \text{Trapdoor}(sk, s_\varepsilon)$: Having input the secret key sk and one keyword ε , it outputs one trapdoor t_ε in this algorithm.

$(1 \text{ or } 0) \leftarrow \text{Verification}(t_\varepsilon, s_\varepsilon)$: With the input of a trapdoor t_ε and a searchable ciphertext s_ε , this algorithm designs to output the comparison decision bit 1 if $\varepsilon t = \varepsilon$, or 0 otherwise.

4. Our Proposed Scheme

4.1. Blockchain Architecture. Blockchain is essentially a decentralized database, which is a string of blocks that are associated using cryptography methods. Each transaction includes hash function, Merkle tree, and so on. In this paper, we replace the search party in searchable encryption with blockchain to ensure the credibility of the search process. As shown in Figure 1, our paper optimizes and adjusts the five-layer architecture of the original blockchain and adds a data retrieval function to the application layer to ensure that the blockchain network can base on the algorithm written in the

smart contract realizing the retrieval of the keyword ciphertext.

4.2. System Model. In this section, we give an introduction to the system model of our PPSEB scheme in Figure 2, with four main entities, including patient, doctor, a cloud server, and blockchain network.

- (1) Patient: The patient integrates Electronic Health Record (EHR), including various medical information such as drug-using records as a patient. Moreover, the patient encrypts the EHR and uploads it to the Cloud Server. Then the patient generates a set of keywords {keywords, sequence number} related to the specified keyword and adds blocks to the blockchain.
- (2) Doctor: The doctor needs to generate a trapdoor to search for information about patients. The doctor submits the corresponding trapdoor to the blockchain.
- (3) Blockchain: After receiving the trapdoor from the doctor, the blockchain network will start chain code retrieval to search the corresponding sequence number and submit it to the CloudServer.
- (4) Cloud Server: After receiving the query request, the Cloud Server can use trapdoor to search for all encrypted data and return the query results of the ciphertext corresponding to the keywords to the doctor. During the entire process, the server is unable to obtain any information about the data and keywords.

4.3. The Scheme of PPSEB. In this section, we present our proposed scheme in detail. There are mainly seven steps of our scheme, including PPSEB.Initialization, PPSEB.KeyExt, PPSEB.Encrypt, PPSEB.PEKS, PPSEB.Trapdoor, PPSEB.Verification, and PPSEB.Decrypt, which are elaborated in the following paragraphs and algorithms.

$(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_\varepsilon) \leftarrow \text{PPSEB.Initialization}(k, X, \delta, \sigma)$: Firstly, we have to input one security parameter k , the discrete Gaussian Distribution X and its parameters $\delta = (\delta_1, \delta_2, \dots, \delta_\eta)$, $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_\eta)$ in one period j , where $j = 1, 2, \dots, \eta$. After that, the initialization step is shown as follows.

$(SK_{r||j}, pk_{r||j}) \leftarrow \text{PPSEB.KeyExt}((X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_\varepsilon), j, sk_{r||i}, i)$: After inputting the set Algorithm 1.

$(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_\varepsilon)$ obtained from the Initialization step, we also have to input the current period j together with the secret key $sk_{r||i}$ in the previous period i . Then, the doctor will procedure the following operations, which shows in Algorithm 2.

$(N, W, I_M) \leftarrow \text{PPSEB.Encrypt}(M, pk_{r||j})$: Firstly, the patient divides the medical data M into groups, named $M = (M_1, M_2, \dots, M_n)$, and generates an index $N = (1, 2, \dots, n)$ for each group. After that, the patient extracts keywords from each group of medical data and records them as $W = (w_1, w_2, \dots, w_n)$. Finally, the patient

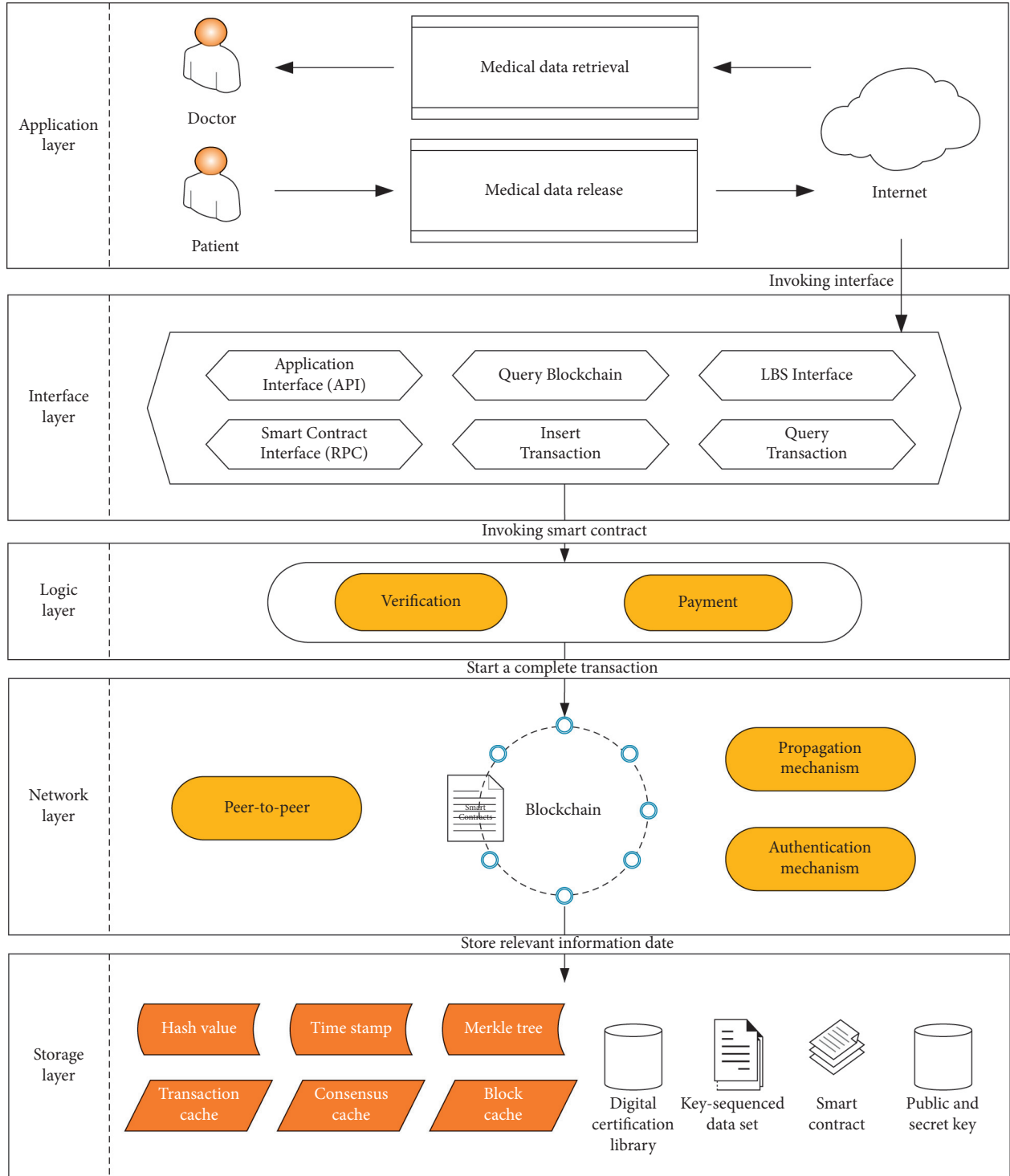


FIGURE 1: Blockchain architecture.

encrypts each group of medical data with the doctor's public key $pk_{r||j}$ at time j , obtains a ciphertext set $CM = (CM_1, CM_2, \dots, CM_n)$, and generates an index set of the medical data ciphertext $I_M = \{(1, CM_1), (2, CM_2), \dots, (n, CM_n)\}$, and it will be stored in the cloud server.

$(CT_j) \leftarrow PPSEB.PEKS((X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s), j, SK_{r||j}, w)$: The patient will procedure $PPSEB.PEKS$ algorithm and input the set $(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s)$, the

public key $pk_{r||j}$, the current time j , and keyword w . This Probabilistic Polynomial Time (PPT) algorithm shows in detail as below. For each keyword $w_i \in W$, the patient executes $PPSEB.PEKS$ algorithm, obtains $CT_W = (CT_{j_1}, CT_{j_2}, \dots, CT_{j_n})$, and pairs each keyword ciphertext with the number to generate keyword index set $I_W = \{(1, CT_{j_1}), (2, CT_{j_2}), \dots, (n, CT_{j_n})\}$. When we get I_W , the patient calculates the hash value H_1 of I with his own

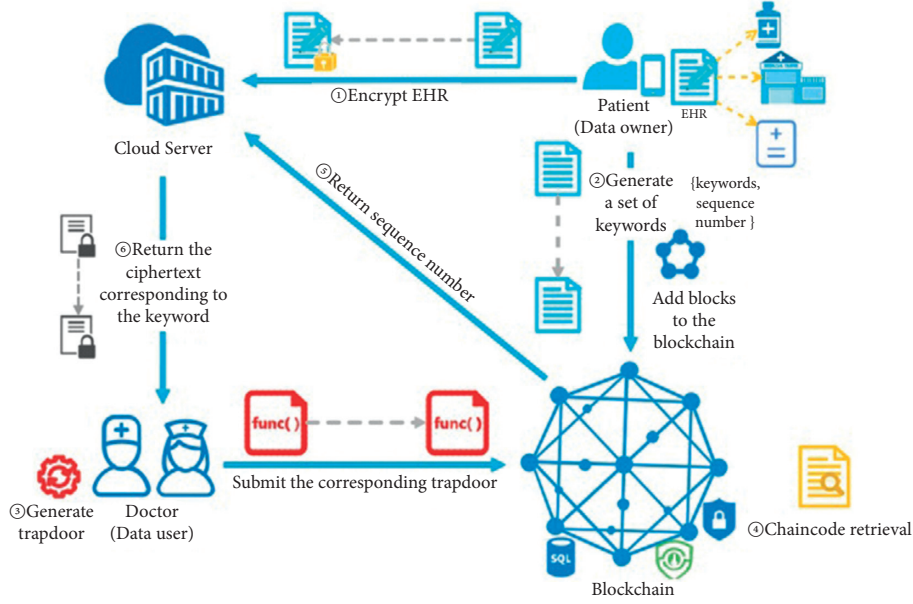


FIGURE 2: System architecture.

Input: security parameter k , discrete Gaussian Distribution X , security Gaussian Distribution δ, σ .

Output: The set $(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s)$

- (1) Select one uniform vector randomly $\mu \leftarrow \mathbb{Z}_q^n$
- (2) Assume that $N = \{0, 1, \dots, \eta\}$ and compute $\mathbb{Z}_q^{n \times m} \times N \rightarrow \mathbb{Z}_q^{m \times m}$ and $\{0, 1\}^{l_1} \times N \rightarrow \mathbb{Z}_q^{m \times m}$
- (3) Set these two hash functions: $H_1: \mathbb{Z}_q^{n \times m} \times N$ and $H_2: \{0, 1\}^{l_1} \times N$
- (4) Call TrapGen(q, n) algorithm to generate $pk_s \in \mathbb{Z}_q^{n \times m}$ and $sk_s \in \mathbb{Z}_q^{m \times m}$, where pk_s and sk_s are public key and secret key of patient, respectively
- (5) Call TrapGen(q, n) algorithm to generate $pk_r \in \mathbb{Z}_q^{n \times m}$ and $sk_r \in \mathbb{Z}_q^{m \times m}$, where pk_r and sk_r are public key and secret key of doctor, respectively
- (6) **Return** the set $(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s)$

ALGORITHM 1: $(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s) \leftarrow PPSEB.Initialization(k, X, \delta, \sigma)$.

Input: set $(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s)$, current time period j , secret key $sk_{r\|i}$ in previous time period i

Output: $SK_{r\|j}$ and $pk_{r\|j}$, where is the secret key during this period j

- (1) Compute $H_1(pk_r\|i)H_1(pk_r\|i-1)\dots H_1(pk_r\|1) \in \mathbb{Z}_q^{m \times m}$
- (2) Set $R_{r\|j} = H_1(pk_r\|i)H_1(pk_r\|i-1)\dots H_1(pk_r\|1)$
- (3) Compute $pk_r(R_{r\|j})^{-1} = pk_r(H_1(pk_r\|i)H_1(pk_r\|i-1)\dots H_1(pk_r\|1))^{-1} \in \mathbb{Z}_q^{n \times m}$
- (4) Set $pk_{r\|i} = pk_r(R_{r\|i})^{-1} = pk_r(H_1(pk_r\|i)H_1(pk_r\|i-1)\dots H_1(pk_r\|1))^{-1}$
- (5) Compute $H_1(pk_r\|j)H_1(pk_r\|j-1)\dots H_1(pk_r\|i+1) \in \mathbb{Z}_q^{m \times m}$
- (6) Set $R_{r\|i \rightarrow j} = H_1(pk_r\|j)H_1(pk_r\|j-1)\dots H_1(pk_r\|i+1)$
- (7) Call NewBasisDel($pk_{r\|i}, R_{r\|i \rightarrow j}, sk_{r\|j}, \delta_j$) to compute $SK_{r\|j} \leftarrow sk_{r\|j}$, where $SK_{r\|j}$ is the secret key during this period j
- (8) Compute $pk_{r\|i}(R_{r\|i \rightarrow j})^{-1} = pk_r(R_{r\|j})^{-1} \in \mathbb{Z}_q^{n \times m}$
- (9) Set $pk_{r\|j} = pk_r(R_{r\|j})^{-1}$
- (10) **Return** $SK_{r\|j}$ and $pk_{r\|j}$

ALGORITHM 2: $(SK_{r\|j}, pk_{r\|j}) \leftarrow PPSEB.KeyExt((X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s), j, sk_{r\|i}, i)$.

private key to generate a digital signature, writes down the transaction $I D$ and timestamp, generates the corresponding transaction, and submits it to the master node for verification. After that, all nodes of the blockchain network execute the consensus algorithm, and the master node jointly

packs the transaction orders in a period of time to form a block and then sends it to the affiliate node. Then, the affiliate node receives the block sent by the master node and verifies the transaction slip contained in the block. Firstly, the affiliate node extracts the public key of the patient stored in the

transaction sheet from the node and decrypts the digital signature and get the hash value H_2 of I_W . If $H_1 = H_2$, the affiliate node declares that the verification is successful. Otherwise, it means that the data may be tampered with and return this transaction to the patient. Assuming that the maximum number of malicious nodes that can exist in the consensus algorithm is f , if the number of verifications passes $Num = f + 1$, the block will be stored in each node of the blockchain network Algorithm 3.

$Trap_{w\parallel j} \leftarrow PPSEB.Trapdoor((X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s), (pk_{r\parallel j}, sk_{r\parallel j}), j, w)$: The doctor will procedure this algorithm after inputting the set $(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s)$, the public key and secret key pair $(pk_{r\parallel j}, sk_{r\parallel j})$ of the medical doctor during this period j , and one keyword $w \in W$. The detailed description is shown in Algorithm 4.

Finally, the doctor will send $Trap_{w\parallel j}$ to the blockchain through an efficient and secure communication channel.

N_0 or False $\leftarrow PPSEB.Verification((X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s), CT_j, t_{w\parallel j})$: This PPT algorithm produced by the blockchain inputs including the set $(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s)$, the ciphertext CT_j , one trapdoor $Trap_{w\parallel j}$ in this period j of the doctor. If it outputs true; it means that the trapdoor $Trap_{w\parallel j}$ and the ciphertext CT_j contain the uniform keyword w . Then, the blockchain returns the number N_0 of the ciphertext corresponding to the keyword to the cloud server. The cloud server finds the ciphertext of the keyword according to N_0 and returns it to the doctor Algorithm 5.

$M_0 \leftarrow PPSEB.Decrypt(CM_0, j, SK_{r\parallel j})$: After the doctor obtains the ciphertext CM_0 of the medical data returned by the cloud server, he/she decrypts it with his $SK_{r\parallel j}$ at time j to obtain the plaintext of medical data M_0 .

5. Security Analysis

In this section, we will demonstrate our scheme's correctness and provable security to achieve the security of the keyword ciphertext in our scheme under random oracle.

5.1. Correctness. In this section, we suppose that the key pair at time j of doctors and patients are $(pk_{r\parallel j}, sk_{r\parallel j})$, $(pk_{s\parallel j}, sk_{s\parallel j})$, respectively. Then, we set w as the keyword of the ciphertext CT_j and then w' is a keyword that matches the trapdoor $Trap_{w'\parallel j}$. It is well known that the cloud server can use $Trap_{w'\parallel j}$ at a time j to recover $(y_{j1}', y_{j2}', \dots, y_{ji}') = CT_{j1} - Trap_{w'\parallel j}^T CT_{j2}$ in $PPSEB.Verification$. Since the relationship between w and w' is uncertain, we divide the discussion into the following two situations:

Case 1: If $w \neq w'$, then $CT_{j1} - Trap_{w'\parallel j}^T CT_{j2} \neq CT_{j1} - Trap_{w\parallel j}^T CT_{j2}$, so we can decrypt the ciphertext CT_j and obtain that: for $i = 1, 2, \dots, l$, there must be $y_{ji} \neq 1$.

Case 2: If $w = w'$, then there is $CT_{j1} - Trap_{w'\parallel j}^T CT_{j2} = CT_{j1} - Trap_{w\parallel j}^T CT_{j2} = noi_j + (y_{j1}, y_{j2}, \dots, y_{ji}) \lfloor q/2 \rfloor - Trap_{w\parallel j}^T CT_{j2}$. Among them, $noi_j - Trap_{w\parallel j}^T CT_{j2}$ is a

noise vector. According to [25], we need to ensure that the error vector is less than $q/5$, so that the decryption process does not make mistakes. Consequently, we can compute that: for $i = 1, 2, \dots, l$, $y_{ji}' = 1$.

So, the cloud server can ensure that the keyword w can correspond to the ciphertext $CT_j = (CT_{j1}, CT_{j2})$ and the trapdoor $Trap_{w\parallel j}^T$; that is, PPSEB can achieve correctness.

Last but not least, the cloud server sends the encrypted medical data corresponding to the keyword w to the doctor, and the doctor obtains the corresponding plaintext data after decrypting it according to its key.

5.2. Provable Security

Theorem 1. *In the PPSEB, the difficulty of the attacker to crack the indistinguishability of the ciphertext can be reduced to the difficulty of the LWE problem.*

Proof. Suppose that there is an attacker A under the random oracle model, which can crack the indistinguishability of the ciphertext in polynomial time. On this basis, we have created a challenger C having the ability to solve the LWE problem. \square

5.2.1. Setup. To begin with, challenger C sends $(u_k, v_{k1}, v_{k2}, \dots, v_{kl})$, $k = 0, 1, \dots, m$ from a random oracle machine. Then, C guesses $\tau = j^*$ as a point in time when A breaks the indistinguishability of the ciphertext. After that, C creates two lists, named L_1 and L_2 . Finally, C interacts with attacker A. The steps are as follows:

- (1) Challenger C runs the SampleR algorithm to obtain R, then C selects $\tau + 1$ vectors from $R^*, R_1^*, \dots, R_\tau^*$ and assembles it into a matrix $F^* \in \mathbb{Z}_q^{n \times m}$, making u_k the k -th column of F^* .
- (2) Challenger C obtains $pk_r = F^* R^* R_1^* \dots R_\tau^*$. Because F^* is independent of $\mathbb{Z}_q^{n \times m}$ and $R_1^*, R_2^*, \dots, R_\tau^*$ are irreversible matrices, pk_r is independent of $\mathbb{Z}_q^{n \times m}$. Then, C selects a matrix as $pk_s \in \mathbb{Z}_q^{n \times m}$ and sets $\mu = u_0 \in \mathbb{Z}_q^n$ to get a set $(pk_r, pk_s, \mu, H_1, H_2)$. Last but not least, C sends $(pk_r, pk_s, \mu, H_1, H_2)$ to attacker A. After receiving the set $(pk_r, pk_s, \mu, H_1, H_2)$, A executes H_1 query and H_2 query.

H_1 query: A initiates an inquiry to each $pk_{r\parallel j}$, where $j = 1, 2, \dots, \tau$. C computes $R_j^* = H_1(pk_{r\parallel j})$ and sends R_j^* to A.

Case 1: $j = \tau + 1$. Challenger C gets $pk_{r\parallel j-1} = pk_r \cdot (R^* R_1^* \dots R_\tau^*)^{-1}$ and runs Sample R with Basis algorithm to get R_j and the basis $sk_{r\parallel j}$ of lattice $L_q^1(A_{r\parallel j})$, where $A_{r\parallel j} = R_j^{-1} \cdot A_{r\parallel j-1}$. Then, C appends $(pk_{r\parallel j}, pk_{r\parallel j}, R_j, sk_{r\parallel j})$ to the list L_1 . Consequently, C transmits R_j to attacker A.

Case 2: $j > \tau + 1$. Challenger C finds $(pk_{r\parallel j-1}, pk_{r\parallel j-1}, R_{j-1}, sk_{r\parallel j-1})$ from the L_1 . Then, C selects a

Input: set $(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s)$, current time period j , secret key $SK_{r||j}$ in current period j
Output: CT_j

- (1) Set a binary string $B_j \leftarrow \mathbb{Z}_q^{n \times l}$, where l is the security level of test in medical data cloud storage
- (2) Select a unitive matrix $B_j \leftarrow \mathbb{Z}_q^{n \times l}$ of $(n \times l)$ dimension
- (3) Select noise $noi_{j1}, noi_{j2}, \dots, noi_{jl} \leftarrow \mathbb{Z}_q$ through X
- (4) Set $noi_j = (noi_{j1}, noi_{j2}, \dots, noi_{jl})$
- (5) Select each noise vector $noiv_{j1}, noiv_{j2}, \dots, noiv_{jl} \leftarrow \mathbb{Z}_q$ on the basis of X^m
- (6) Set the noise vector matrix $noiv_j = (noiv_{j1}, noiv_{j2}, \dots, noiv_{jl}) \in \mathbb{Z}_q^{m \times l}$
- (7) Assume $\beta_j = H_2(w||j)$ and then compute $CT_{j1} = \mu^T B_j + noi_j + y_j \lfloor q/2 \rfloor$ and $CT_{j2} = (pk_{r||j} \beta_j^{-1})^T B_j + noiv_j$ as ciphertext
- (8) Set ciphertext $CT_j = (CT_{j1}, CT_{j2}) = (\mu^T B_j + noi_j + y_j \lfloor q/2 \rfloor, (pk_{r||j} \beta_j^{-1})^T B_j + noiv_j)$
- (9) **Return** CT_j to doctor

ALGORITHM 3: $(CT_j) \leftarrow PPSEB.PEKS((X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s), j, SK_{r||j}, w)$.

Input: set $(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s)$, current period j , public-secret key pair $(pk_{r||j}, sk_{r||j})$, one keyword w
Output: $sk_{w||j}$ and $Trap_{w||j}$

- (1) Compute $\beta_j = H_2(w||j)$
- (2) Set $R_{r||j} = H_1(pk_{r||j}) H_1(pk_{r||j} - 1) \dots H_1(pk_{r||j} - l)$
- (3) **Call** NewBasisDel $(pk_{r||j}, \beta_j, sk_{r||j}, \delta_j)$ to generate one short lattice basis $sk_{w||j} \in \mathbb{Z}_q^{m \times m}$ in random
- (4) **Call** SamplePre $(pk_{r||j} \beta_j^{-1}, sk_{w||j}, \mu, \sigma_j)$ to generate the trapdoor $Trap_{w||j} \in \mathbb{Z}_q^m$
- (5) **Return** $Trap_{w||j}$

ALGORITHM 4: $Trap_{w||j} \leftarrow PPSEB.Trapdoor((X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s), (pk_{r||j}, sk_{r||j}), j, w)$.

Input: set $(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s)$, ciphertext CT_j , current period j , trapdoor $Trap_{w||j}$
Output: N_0 or False

- (1) Compute $(y_{j1}, y_{j2}, \dots, y_{jl}) = CT_{j1} - Trap_{w||j}^T CT_{j2}$
- (2) Set $y_j = (y_{j1}, y_{j2}, \dots, y_{jl})$
- (3) Select integer q satisfies $\{1, 2, \dots, q\} \subset \mathbb{Z}^+$
- (4) **for** $(i = 1, 2, \dots, l)$ **do**
- (5) **if** $|y_{ji} - \lfloor q/2 \rfloor| \geq \lfloor q/4 \rfloor$ **then**
- (6) The medical cloud sever will abort it and Return False.
- (7) **else**
- (8) Set $y_{ji} = 1$ up to $y_{ji} = 1$
- (9) **end if**
- (10) **endfor**
- (11) **if** $y_j = (1, 1, \dots, 1) \in \{1\}^l$ **then**
- (12) Return N_0
- (13) **else**
- (14) Return False
- (15) **end if**

ALGORITHM 5: $True \text{ or } False \leftarrow PPSEB.Verification((X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s), CT_j, t_{w||j})$.

matrix R_j , and carries out the New Basis Del algorithm to compute $sk_{r||j}$ as the basis of $L_q^\perp(pk_{r||j})$, where $pk_{r||j} = pk_{r||j-1} \cdot R_j^{-1}$. Consequently, C appends $(pk_{r||j}, pk_{r||j}, R_j, sk_{r||j})$ to L_1 , and transmits R_j to attacker A.

H_2 query: The attacker A queries w , at the same time challenger C performs the following operations:

Case 1: $w = w^*$ and $j = j^*$. The challenger C calculates $R^* = H_2(w||j)$ and sends R^* to A.

Case 2: $w \neq w^*$ or $j \neq j^*$. The challenger C looks for $(pk_{r||j}, pk_{r||j}, R_j, sk_{r||j})$ in L_1 , selects a matrix $R_{w||j}$, and executes the NewBasisDel algorithm to generate a basis $sk_{w||j}$ of $L_q^\perp(pk_{r||j} \cdot R_{w||j}^{-1})$. Finally, C saves $(w||j, pk_{r||j} \cdot R_{w||j}^{-1}, R_{w||j}, sk_{w||j})$ in L_2 , and sends $R_{w||j}$ to A.

5.2.2. *Trapdoor Query.* When C receives a query for a keyword w from A, C first looks at L_2 , and if there is no $(w||j, pk_{r||j} \cdot R_{w||j}^{-1}, R_{w||j}, sk_{w||j})$ in L_2 ; then this process will be restarted.

Otherwise, C gets the private key $sk_{w_{\parallel j}}$, runs the SamplePre algorithm to generate a trapdoor $Trap_{w_{\parallel j}}$, and sends it to A.

5.2.3. Break-In Phase. In this process, attacker A can query the private key of the doctor in the $j > j^*$ period, and $j^* = \tau$ is set a break-in time. After A queries H_1 on $pk_r \parallel j$, C sends the private key $sk_r \parallel j$ to A.

In time i , which is the prior period, we can find $(pk_r \parallel j, pk_r \parallel i, R_i, sk_r \parallel i)$ from L_1 because the attacker A will perform H_1 queries on $pk_r \parallel i$. Further, we calculate $pk_r \parallel i = pk_r \parallel \tau + 1 = pk_r \cdot (R_2^* \cdots R_2^* R_1^*)^{-1} \cdot H_1(pk_r \parallel \tau + 1)^{-1}$, which $sk_r \parallel i$ is the basis of the lattice $L_q^\perp(pk_r \parallel i)$. After that, challenger C calculates $R_{r \parallel i \rightarrow j} = H_1(pk_r \parallel j) \cdots H_1(pk_r \parallel i \rightarrow 1)$ and runs the NewBasisDel algorithm to obtain $pk_r \parallel j = pk_r \parallel i \cdot R_{r \parallel i \rightarrow j}^{-1}$ and $sk_r \parallel j$ in time j . Consequently, C sends $sk_r \parallel j$ to attacker A.

5.2.4. Challenge Phase. Assuming that w_0^* and w_1^* are two keywords, challenger C randomly selects a quantity from $\{0, 1\}$ and assigns it to b . Then we need to divide into the following cases according to the value of b .

Case 1: $b = 0$. The challenger C sends ciphertext $(CT_{\tau_1}^*, CT_{\tau_2}^*)$ of w_0^* to A.

Case 2: $b = 1$. We create $v_0 = (v_{01}, v_{02}, \dots, v_{0l})$, $v^* = (v_1, v_2, \dots, v_m)^T$, and $y_j^* = (1, 1, \dots, 1)$. Then, $CT_{\tau_1}^* = v_0 + \lfloor q/2 \rfloor \cdot y_j^*$ and $CT_{\tau_2}^* = v^*$ can be obtained. Consequently, C sends the ciphertext $(CT_{\tau_1}^*, CT_{\tau_2}^*)$ of w_1^* to A.

5.2.5. Guess Phase. In this process, attacker A outputs $b' = 0$ or $b' = 1$ as the response of the Challenge phase.

Analysis: To begin with, according to the basic probability knowledge, the probability of C outputting the ciphertext of the keyword w_1 is $1/2$.

Suppose that A can break the indistinguishability of the ciphertext with the probability p . In addition, the probability that challenger C can correctly obtain the break time is $1/m$. Consequently, C can solve the LWE hardness with the probability of $p/2m$. In a nutshell, the difficulty of the attacker to crack the indistinguishability of the ciphertext can be reduced to the difficulty of the LWE hardness.

6. Performance Evaluation

In this section, to guarantee the forward security, anti-quantum KGA, and suitability in the medical scenarios of our PPSEB scheme, we analyze the computational expense, security property, and network communication costs of our scheme and compare our scheme with existing PEKS schemes [3, 5, 28, 29] on the actual performance in the medical background through experiments and numerical simulation technique. The experiments evaluating and testing the actual performance of our scheme are operated on a MacOS with an Intel Core i7 CPU and 16 GB RAM. The implementation of schemes is based on the C++ language,

and we use medical data extremely close to actual applications of daily life to complete the experiments. Meanwhile, in order to realize the security of the q -ary lattices, the parameters satisfy $m > 2n \log q$, $q \geq 3$, since the algorithms counting on lattice-based cryptography are relied on q, m, n . The notations of the following specific descriptions in the experiments are provided in Table 1. The accurate experimental data of 200 trials on average are shown in the following figures, and the results accord with our design objective extremely.

Our PPSEB is highly efficient compared with other PEKS schemes. As is illustrated in Table 2, the theoretical communication costs of each scheme are listed accurately.

We prove the theoretical value, and the experimental result reflects in Figure 3, demonstrating that the trapdoor size of the PPSEB scheme is the least one among the whole schemes. Along with the stabilizing growth in communication costs, our algorithm is superior to the others, indicating a hidden potential to reduce network resource consumption.

As to the actual performance, Figure 3 indicates that the PPSEB scheme reveals a considerable efficiency advantage. The PEKS size of PPSEB is relatively close to the scheme [3, 5, 28] and much less than the scheme [29]. The trapdoor size in our scheme is a quarter of [29]. However, in terms of postquantum, our proposed PPSEB is more secure than the scheme [3, 5, 28] while being applied in medical data encryption protection. Thus, it is pretty sound and acceptable for PPSEB to increase the nominal communication costs corresponding to PEKS size.

In addition, we not only analyze the computational expense and security property of our scheme but also compare it with existing PEKS schemes [3, 5, 28] through experimental medical data. As shown in Figure 4, the testing time of our scheme is also much shorter than the other existing PEKS schemes. Significantly, the more the number of retrieving keywords increases, the more apparent the superiority becomes.

Besides, we test the testing time and computational expense of the PEKS schemes and record the results in Table 3.

Our scheme realizes nearly the same as a scheme [3] in saving the computational expense and searching efficiency according to the comparison in Figure 5. When the number of retrieving keywords is 180, the testing time of [5] is 7.2s, and ours is 0.477s, which is 15.09 times that of PPSEB. As a result, our scheme is not only advantageous in terms of postquantum property, but also relatively efficient than the other schemes. Consequently, although the introduction of blockchain technology has brought a certain amount of complexity and extra overhead to our system, it is certified that our PPSEB scheme can realize the property of postquantum, forward security on maintaining the confidentiality of medical data and superiority in the applications of medical scenarios. From a more practical view, it is both convenient and swift for doctors to master the patient's physical condition, obtain the patient's medical records, and make the correct diagnosis promptly in practical medical scenarios. In

TABLE 1: Notations of descriptions.

Notations	Descriptions
$Time_{me}$	The modular exponentiation time
$Time_{sm}$	The scalar multiplication time
$Time_{hp}$	The hash-to-point time
$Time_{pa}$	The point addition time
$Time_{bp}$	The bilinear pairing time
$Time_{hf}$	The hash function time
$Time_m$	The multiplication time
S_1	One element bit size in G_1
S_T	One element bit size in G_T
S_p	One element bit size in \mathbb{Z}_p
S_q	One element bit size in \mathbb{Z}_q
S_l	The security level with a value of 10

TABLE 2: Communication costs.

Schemes	Size of PEKS algorithm	Size of trapdoor algorithm
Our scheme	$(S_l + mS_l)S_q$	mS_q
Boneh et al. [3]	$S_p + S_1$	S_1
Ma et al. [5]	$5S_1 + 3S_T$	$3S_1$
Ma et al. [28]	$S_p + S_1$	S_1
Shao et al. [29]	$S_l + S_1$	S_1

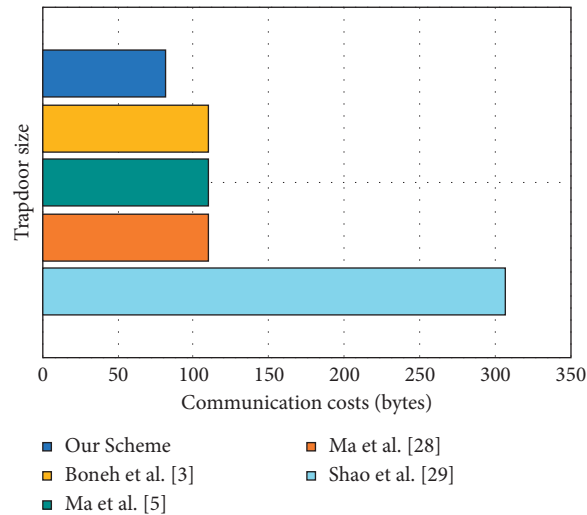


FIGURE 3: Communication costs comparison corresponding to trapdoor size.

addition, the more profound performance of PPSEB on managing medical data of Electronic Health Records systems, such as electronic medical record and electronic prescription, need to be tested experimentally and further study in development.

In Figure 6, we compared the PEKS computational expense of PPESB with [3, 5, 28, 29]. Among them, the PEKS computational expense of our scheme is much smaller than other schemes, which shows that our scheme has higher efficiency under the same number of retrieving keywords.

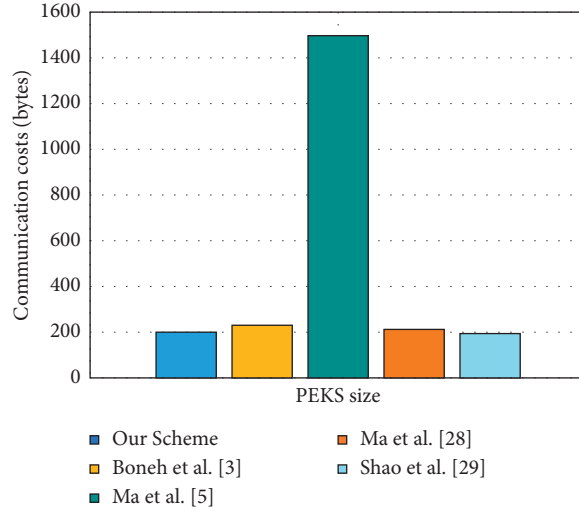


FIGURE 4: Communication costs comparison corresponding to PEKS size.

TABLE 3: Testing time and computational expense.

Schemes	Testing time	PEKS computational expense
Our scheme	$mS_l Time_m$	$(mnS_l + nS_l + m^2n)Time_m + Time_{hf}$
Boneh et al. [3]	$Time_{hf} + Time_{bp}$	$2Time_{pa} + Time_{hp} + 4Time_{sm} + Time_{bp} + 3Time_{hf}$
Ma et al. [5]	$5Time_{me} + 4Time_{bp} + Time_{hf}$	$3Time_{hf} + 9Time_{me} + 3Time_{bp}$
Ma et al. [28]	$Time_{hf} + Time_{bp} + Time_{sm} + 2Time_{pa} + 2Time_{hp}$	$Time_{hf} + 3Time_{bp} + 2Time_m + Time_{pa} + 4Time_{sm} + 3Time_{hp}$
Shao et al. [29]	$Time_{hf} + Time_{bp}$	$2Time_{hf} + 2Time_{me} + Time_{bp}$

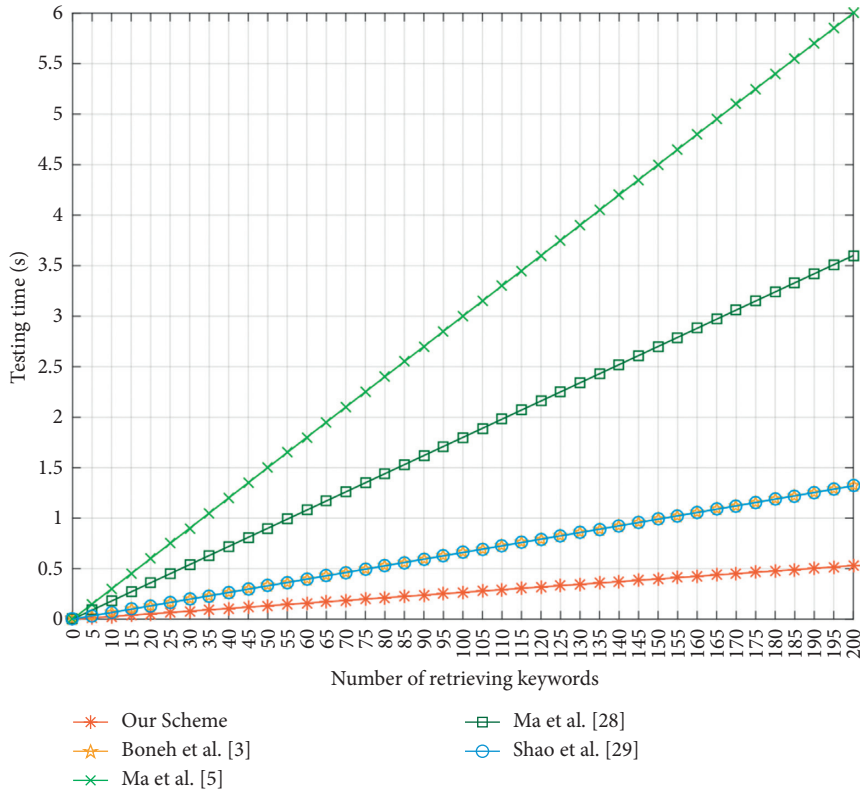


FIGURE 5: The testing time comparison.

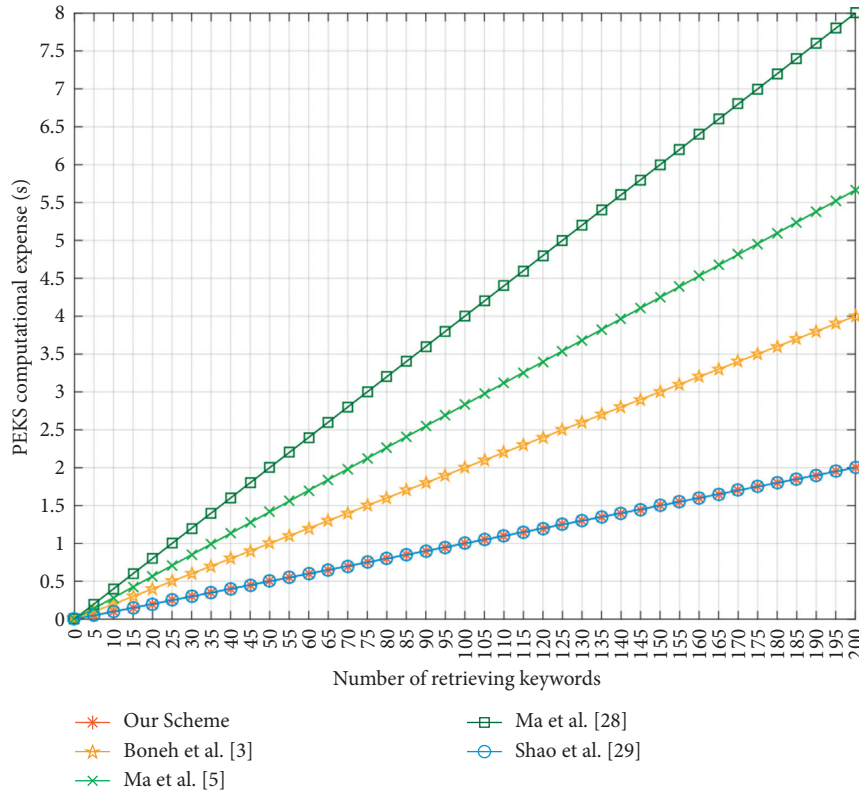


FIGURE 6: PEKS computational expense comparison.

7. Conclusion

In our paper, we proposed postquantum Public-key Searchable Encryption on Blockchain (PPSEB) for E-healthcare scenarios. PPSEB is capable of resisting keyword-guessing quantum computing attacks. Moreover, our proposed scheme combines public-key searchable encryption and blockchain, avoiding turning over the searching process to a third party and enhancing the security level. Furthermore, we assure forward security, maintaining the confidentiality of medical data. Both security analysis and comprehensive performance evaluation demonstrate that PPSEB can achieve the property of searching efficiency and lightweight of lower computational cost in retrieving keywords and generating trapdoor compared with other existing E-healthcare schemes.

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 Open Fund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province (Grant No. SKLACSS-202101), NSFC

(Grant nos. 62176273, 61962009, U1936216, and 62076042), the Foundation of Guizhou Provincial Key Laboratory of Public Big Data (Nos. 2019BDKFJJ010 and 2019BDKFJJ014), the Fundamental Research Funds for Beijing Municipal Commission of Education, Beijing Urban Governance Research Base of North China University of Technology, the Natural Science Foundation of Inner Mongolia (2021MS06006), Baotou Kundulun District Science and technology plan project (YF2020013), and Inner Mongolia discipline inspection and supervision big data laboratory open project fund (IMDBD2020020).

References

- [1] I. M. Baytas, K. Lin, F. Wang, A. K. Jain, J. Zhou, and J. Zhou, "PhenoTree: interactive visual analytics for hierarchical phenotyping from large-scale electronic health records," *IEEE Transactions on Multimedia*, vol. 18, no. 11, pp. 2257–2270, 2016.
- [2] Y. Chen, S. Dong, T. Li, Y. Wang, and H. Zhou, "Dynamic multi-key FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5239–5249, 2021.
- [3] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," vol. 3027, pp. 506–522, in *Proceedings of the 23th Annual International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT 2004)*, vol. 3027, pp. 506–522, Springer, Berlin, Heidelberg, May 2004.
- [4] Q. Huang and H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," *Information Sciences*, vol. 403–404, pp. 1–14, 2017.

- [5] M. Ma, D. He, N. Kumar et al., "Certificateless searchable public key encryption scheme for industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 759–767, 2017.
- [6] R. Chen, Y. Mu, G. Yang et al., "Dual-server public-key encryption with keyword search for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 789–798, 2016.
- [7] T. D. Ladd, F. Jelezko, R. Laflamme et al., "Quantum computers," *Nature*, vol. 464, no. 7285, pp. 45–53, 2012.
- [8] A. Galindo and M. A. Martín-Delgado, "Information and computation: classical and quantum aspects," *Reviews of Modern Physics*, vol. 74, no. 2, pp. 347–423, 2002.
- [9] W. Li, X. Li, J. Gao, and H. Wang, "Design of secure authenticated key management protocol for cloud computing environments," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1276–1290, 2021.
- [10] E. Uchiteleva, A. R. Hussein, A. Shami, and A. Shami, "Lightweight dynamic group rekeying for low-power wireless networks in IIoT," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4972–4986, 2020.
- [11] Y. Cheng, S. Xu, M. Zang et al., "LPPA: a lightweight privacy-preserving authentication scheme for the Internet of drones," in *Proceedings of the 21st International Conference on Communication Technology (ICCT 2021)*, pp. 656–661, IEEE, Tianjin, China, October 2021.
- [12] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
- [13] T. Li, Z. Wang, Y. Chen et al., "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems, early access*, vol. 33, no. 1, 2021.
- [14] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1204–1221, 2022.
- [15] X. Liu, R. Zhang, G. Xu, X.-B. Chen, and N. N. Xiong, "Confidentially judging the relationship between an integer and an interval against malicious adversaries and its applications," *Computer Communications*, vol. 180, pp. 115–125, 2021.
- [16] G. Xu, Y. Cao, S. Xu et al., "A novel post-quantum blind signature for log system in blockchain," *Computer Systems Science and Engineering*, vol. 41, no. 3, pp. 945–958, 2022.
- [17] T. Li, Y. Chen, Y. Wang et al., "Rational protocols and attacks in blockchain system," *Security and Communication Networks*, vol. 2020, pp. 1–11, Article ID 8839047, 2020.
- [18] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008, <https://bitcoin.org/bitcoin.pdf>.
- [19] S. Xu, X. Chen, and Y. He, "EVchain: an anonymous blockchain-based system for charging-connected electric vehicles," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 845–856, 2021.
- [20] B. Chen, L. Wu, H. Wang, L. Zhou, and D. He, "A blockchain-based searchable public-key encryption with forward and backward privacy for cloud-assisted vehicular social networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5813–5825, 2020.
- [21] C. Liu, Y. Xiao, V. Javangula, Q. Hu, S. Wang, and X. Cheng, "NormaChain: a blockchain-based normalized autonomous transaction settlement system for IoT-based E-commerce," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4680–4693, 2019.
- [22] D. Micciancio and O. Regev, "Lattice-based cryptography," vol. 4117, pp. 131–141, in *Proceedings of the 26th Annual International Cryptology Conference (CRYPTO 2006)*, vol. 4117, pp. 131–141, Springer, Berlin, Heidelberg, May 2006.
- [23] S. Xu, X. Chen, C. Wang et al., "A lattice-based ring signature scheme to secure automated valet parking," vol. 12938, pp. 70–83, in *Proceedings of the International Conference on 16th Wireless Algorithms, Systems, and Applications (WASA 2021)*, vol. 12938, pp. 70–83, Springer, Nanjing, China, September 2021.
- [24] J. Alwen and C. Peikert, "Generating shorter bases for hard random lattices," *Theory of Computing Systems*, vol. 48, no. 3, pp. 535–553, 2011.
- [25] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," vol. 14, pp. 197–206, in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC 2008)*, vol. 14, pp. 197–206, ACM, Victoria, British Columbia, May 2008.
- [26] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (H)IBE in the standard model," vol. 6110, pp. 553–572, in *Proceedings of the 29th Annual International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT 2010)*, vol. 6110, pp. 553–572, Springer, Riviera, France, June 2010.
- [27] S. Agrawal, D. Boneh, and X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE," vol. 6223, pp. 98–115, in *Proceedings of the 30th Annual International Cryptology Conference (CRYPTO 2010)*, vol. 6223, pp. 98–115, Springer, Santa Barbara, CA, USA, May 2010.
- [28] M. Ma, D. He, M. K. Khanand, and J. Chen, "Certificateless searchable public key encryption scheme for mobile healthcare system," *Computers & Electrical Engineering*, vol. 65, pp. 413–424, 2017.
- [29] Z.-Y. Shao, B. Yang, and B. Yang, "On security against the server in designated tester public key encryption with keyword search," *Information Processing Letters*, vol. 115, no. 12, pp. 957–961, 2015.

Research Article

Research on the Millionaires' Problem under the Malicious Model Based on the Elliptic Curve Cryptography

Xin Liu,¹ Yang Xu,¹ Gang Xu ,^{2,3} and Baoshan Li¹

¹School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, China

²School of Information Science and Technology, North China University of Technology, Beijing 100144, China

³Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100144, China

Correspondence should be addressed to Gang Xu; gx@ncut.edu.cn

Received 22 October 2021; Accepted 10 February 2022; Published 23 March 2022

Academic Editor: Xin-Yi Huang

Copyright © 2022 Xin Liu 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 rapid development of blockchain, big data, cloud computing, and artificial intelligence, the security of multisource data collaborative computing has become increasingly prominent. Secure multiparty computing has become the core technology of privacy collaborative computing. Millionaires' problem is the cornerstone of secure multiparty computation. Firstly, this paper proposes a 0-1 coding rule, which is used to solve the millionaires' problem under the semihonest model. Aiming at the possible malicious behaviors of the protocol under the semihonest model, the millionaires' problem protocol under the malicious model based on the elliptic curve cryptography is designed by using cryptographic tools such as the zero-knowledge proof and the cut-choose method. This protocol not only can effectively solve the millionaires' problem but also can safely and effectively prevent malicious behaviors. Meanwhile, the security ordering designed by the protocol can be effectively applied to a quality evaluation in the blockchain.

1. Introduction

Secure multiparty computation (SMC) is the core technology to achieve collaborative computing for the privacy of multisource data in recent years. The idea of SMC is proposed by Professor Yao Qizhi in 1982 [1], and then Goldreich [2, 3] began to do more in-depth research on SMC. SMC has been widely used in the blockchain [4–6], data mining [7–9], privacy computing [10, 11], medical [12, 13].

The millionaires' problem is one of the most classic problems in SMC. Many cryptographers have been working on it. Reference [14] presents a protocol for solving the millionaires' problem based on the exchange cryptosystem, oblivious transfer method. Based on the Goldwasser–Micali (GM) cryptography, reference [15] proposed a protocol to solve the problem of socialist millionaires. Reference [16] proposed a protocol to solve the problem of the millionaires' problem by using the shift registers and the property of probability encryption. Reference [17] presents a protocol

based on the Paillier cryptosystem. The existing schemes have the disadvantage of inefficiency, and most of them are only suitable for the semihonest model and cannot resist malicious attacks. To solve the above problems, this paper studies the millionaires' problem under the malicious model in depth and presents the millionaires' problem protocol based on the elliptic curve cryptography.

Elliptic curve cryptography (ECC) is a traditional encryption method that has the advantage of high computational efficiency and is based on the elliptic curve discrete logarithm problem, and it has been widely used because of its short key [18–20]. Using ECC, the protocol designed in this paper has more efficient operation efficiency and security. The main contributions are as follows:

- (1) First, a 0-1 encoding rule for ECC is proposed, and then a millionaires' problem protocol under the semihonest model is designed
- (2) With the help of some cryptographic tools such as the zero-knowledge proof and cut-choose method, a

protocol is designed to resist the attacks of malicious opponents

- (3) Finally, an ideal-practical example method is used to prove the security of the protocol under the malicious model

2. Preliminary Knowledge

2.1. Elliptic Curve Cryptography. Elliptic curve cryptography (ECC) is a public-key cryptosystem based on discrete logarithmic problems of point groups of elliptic curves. For example, an elliptic curve is defined as $y^2 = x^3 - x$, two points P and Q on the curve, a straight line through P and Q , an intersecting elliptic curve at R' point, and a line perpendicular to X axis through R' point. An intersecting elliptic curve at another point R is defined as $P + Q = R$, as shown in Figure 1.

When $P = Q$, the tangent of the point P intersects R' , and then the point R' makes a straight line perpendicular to the X -axis, intersecting the elliptic curve at another point R . When k identical P are added, they are counted as kP , such as $P + P + P + P = P + 3P = 4P$. Elliptic curves make use of the mathematical problem of discrete logarithm in the above operations, that is, when $K = kP$, P and K are known, and K is easily obtained. But it is very difficult to find P and K when K is known.

In cryptography systems, if there is an elliptic curve $E_p(a, b)$ and a base point G , a random number k is generated as the private key, and the datum is computed k times to get the public key $K = kG$. In ECC, the private key k is easy to get the public key K ; the public key K cannot get the private key k .

Compared with the traditional public key algorithms, elliptic curve cryptography has the following advantages [19, 20]:

- (1) higher security performance. For example, 160 bit ECC has the same security strength as 1024 bit RSA and DSA algorithms.
- (2) small amount of computation and fast processing speed. ECC is much faster than RSA and DSA in the processing speed of private key (decryption and signature).
- (3) The storage space occupation is small. Compared with RSA and DSA, the key size and system parameters of ECC are much smaller, so the storage space occupation is much smaller.
- (4) ECC has a wide application prospect because of its low bandwidth requirements.

2.1.1. ECC Encryption

- (1) Elliptic curve $E_p(a, b)$, a base point G , private key k , and public key $K = kG$
- (2) Encode a plaintext a to a point M on the elliptic curve $E_p(a, b)$ and select a random number $r < n$ (where n is the order of base point G)

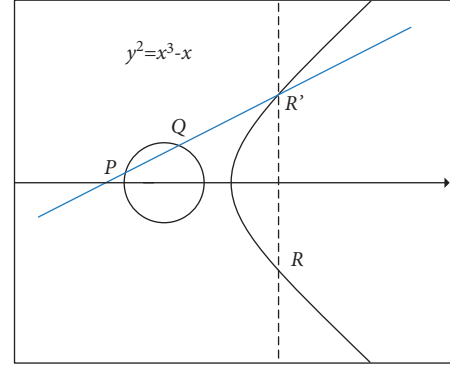


FIGURE 1: Elliptic curve operation, defined $P + Q = R$.

- (3) Encryption: $C_1 = M + rK$, $C_2 = rG$
- (4) Decryption: $C_1 - kC_2 = M + rK - k(rG) = M + r(K - kG) = M$

2.1.2. Additive Homomorphism of ECC

- (1) Encryption: encode a plaintext a_i ($0 < i \leq m$) onto a point M_i of the elliptic curve $E_p(a, b)$, use the private key k to generate the public key $K = kG$, select a random number $r_i < n$, and calculate $C_{1i} = M_i + r_iK$ and $C_{2i} = r_iG$
- (2) Addition operation: all ciphertexts are added to obtain ciphertexts $(\sum_{i=1}^m C_{1i}, \sum_{i=1}^m C_{2i})$, where $\sum_{i=1}^m C_{1i} = C_{11} + C_{12} + \dots + C_{1m}$ and $\sum_{i=1}^m C_{2i} = C_{21} + C_{22} + \dots + C_{2m}$
- (3) Decryption: compute $\sum_{i=1}^m C_{1i} - k \sum_{i=1}^m C_{2i} = \sum_{i=1}^m M_i + K \sum_{i=1}^m r_i - kG \sum_{i=1}^m r_i = \sum_{i=1}^m M_i$ and finally decode $\sum_{i=1}^m M_i$ to obtain the plaintext $\sum_{i=1}^m a_i$

2.2. Zero-Knowledge Proof. Zero-knowledge proof [21] means that the prover can make the verifier believe that a conclusion is correct without providing any useful information to the verifier. The prover proves to the verifier and makes him believe that he knows or owns a certain message, but the certification process cannot disclose any information about the proved message to the verifier. A large number of facts have proved that the zero-knowledge proof is very useful in cryptography. If the zero-knowledge proof can be used for verification, many problems can be effectively solved.

2.3. Cut-Choose Method. In cryptography, we encrypt the information that can be sent into n messages using n different random numbers. The receiver selects $n/2$ of them to verify their correctness and then selects one of the remaining $n/2$ for the remaining protocol steps. The cut-choose method can minimize the malicious input probability of the protocol and make the transmission of information more secure [22].

2.4. Security Definition of a Protocol under the Malicious Model. To prove that a protocol is secure under the malicious model, it must satisfy the security definition under the

malicious model. If the actual protocol achieves the same security as the ideal protocol, then the protocol is secure [3].

Alice owns x , and Bob owns y . They compare by calculating function $f(x, y) = (f_1(x, y), f_2(x, y))$ with a trusted third party (TTP). At the end of the protocol, both parties get $f_1(x, y)$ and $f_2(x, y)$ without leaking x and y . The ideal model is as follows:

- (1) The honest participant always provides x or y to the TTP, while the malicious participant may decide not to execute the protocol based on x or y or provide a false input x' or y' to the TTP when the protocol is executed.
- (2) TTP sends the result to Alice. After TTP gets input pair (x, y) , $f(x, y)$ is calculated, and $f_1(x, y)$ is sent to Alice; otherwise, the special symbols \perp are sent to Alice.
- (3) TTP sends the result to Bob, and if Alice is a malicious participant, it may no longer contact with TTP after receiving $f_1(x, y)$. In this case, TTP sends

Bob a special symbol \perp ; otherwise, TTP sends $f_2(x, y)$ to Bob.

The ideal protocol is the safest protocol because participants cannot get any information except their $f_i(x, y)$ from TTP. If an actual protocol achieves the same security as the ideal protocol, we say that the actual protocol is secure.

If the participant in the ideal model has an auxiliary information z and the process of calculating $F(x, y)$ in combination with policy \bar{B} is $\text{IDEAL}_{F, \bar{B}(z)}(x, y)$, it is defined as the adversary evenly choosing a random number r and to make $\text{IDEAL}_{F, \bar{B}(z)}(x, y) = \gamma(x, y, z, r)$, where $\gamma(x, y, z, r)$ is defined as follows (note: if both parties under the malicious model are malicious, it is impossible to design an SMC protocol; we do not consider this case):

- (1) If Alice is honest, there is $\gamma(x, y, z, r) = (f_1(x, y'), B_2(y, z, r, f_2(x, y')))$, where $y' = B_2(y, z, r)$.
- (2) If Bob is honest:

$$\gamma(x, y, z, r) = \begin{cases} (B_1(x, z, r, f_1(x', y)), \perp), \perp, & \text{if } : B_1(x, z, r, f_1(x', y)) = \perp, \\ (B_1(x, z, r, f_1(x', y)), f_2(x', y)), & \text{otherwise.} \end{cases} \quad (1)$$

In both cases, $x' = B_1(x, z, r)$.

Definition 1. Security for the malicious model.

If, in the ideal model, an acceptable policy pair $\bar{A} = (A_1, A_2)$ in the actual protocol can be found, there is an acceptable policy pair $\bar{B} = (B_1, B_2)$, such that

$$\{\text{IDEAL}_{F, \bar{B}(z)}(x, y)\}_{x, y, z} \stackrel{c}{=} \{\text{REAL}_{\bar{A}(z)}(x, y)\}_{x, y, z}. \quad (2)$$

So this protocol can calculate F securely.

3. The Protocol under the Semihonest Model

3.1. Solution Ideas. Alice owns x , and Bob owns y . Alice and Bob want to compare the relationship: $x > y$, $x = y$, $x < y$, while they do not want to leak their x and y , respectively. The solution is to code x and y as a set consisting of 1 and 0 and use ECC to design an efficient protocol.

The 0-1 encoding rule: encode x into a set $X = (a_1, a_2, \dots, a_m)$, where $a_1 < a_2 < \dots < a_m$ and

$$a_i = \begin{cases} 1, & i = x; \\ 0, & i \neq x. \end{cases} \quad (3)$$

The comparison rule: based on the position of y in X : if $x > y$, then $\sum_{i=1}^{y-1} a_i + \sum_{i=y}^m a_i = 0$; if $x = y$, then $\sum_{i=1}^{y-1} a_i + \sum_{i=y}^m a_i = 1$; and if $x < y$, then $\sum_{i=1}^{y-1} a_i + \sum_{i=y}^m a_i = 2$. Define the following formula to judge the relationship between x and y :

$$P(x, y) = \begin{cases} 0, & x > y; \\ 1 & x = y; \\ 2, & x < y. \end{cases} \quad (4)$$

For example: Alice's data is 5, which is encoded into a new set $X = (0, 0, 0, 0, 1, 0, 0)$, and Bob calculates with three different data $y = 2, 5, 7$. This is shown in Table 1.

3.2. Specific Protocol. Alice owns data x , and Bob owns data y , and both parties compute $P(x, y)$ securely to determine the relationship. Using the above 0-1 encoding rule, Algorithm 1 under the semihonest model is designed based on ECC homologous encryption.

Algorithm 1 is secure under the semihonest model, but if one of Alice and Bob is a malicious participant, the protocol is no longer secure. The following section will improve the protocol to make it safe and feasible under the malicious model.

4. The Protocol under the Malicious Model

4.1. Solution Ideas. Firstly, we analyze the possible malicious attacks in Algorithm 1. Then the solutions to these malicious attacks are proposed. Finally, the possible malicious attacks cannot be implemented or found when they are committed. The following malicious attacks may exist in Algorithm 1 as follows:

- (1) In Algorithm 1, Alice has both the public key K and the private key k , but Bob only has the public key K , so the final result can only be calculated unilaterally

TABLE 1: The 0-1 encoding. Data comparison results of Alice and Bob.

Alice's data	New set for encoding	Bob's data	Calculate $w = \sum_{i=1}^{y-1} a_i + \sum_{i=1}^y a_i$	Comparison results
		2	$w = 0 + 0 = 0$	$x > y$
5	$X = (0, 0, 0, 0, 1, 0, 0)$	5	$w = 0 + 1 = 1$	$x = y$
		7	$w = 1 + 1 = 2$	$x < y$

Input: Alice owns data x , and Bob owns data y .

Output: $P(x, y)$.

- (1) Alice encodes x into a set $X = (a_1, a_2, \dots, a_m)$, where $a_1 < a_2 < \dots < a_m$ and $a_i = \begin{cases} 1, & i = k; \\ 0, & i \neq k. \end{cases}$
- (2) Alice chooses an elliptic curve E_p , the base point G , and the private key k ; then calculates $kG = K$ as the public key K ; and publishes the public key K and the base point G .
- (3) Alice encodes the plaintext $X = (a_1, a_2, \dots, a_m)$ one by one onto point M_i ($1 \leq i \leq m$) on the elliptic curve E_p (the encoding method is not unique [18], which is not discussed here). She chooses m random numbers r_i and encrypts each element M_i one by one using the public key K of ECC, that is, $E(M_i) = (C_{1i}, C_{2i})$, $C_{1i} = M_i + r_i K + C_{2i} = r_i G$. She gets $E(X) = (E(M_1), E(M_2), \dots, E(M_m))$, which is sent to Bob.
- (4) Bob calculates $E(W) = (C_1, C_2)$ based on the y position of data in $E(X)$, where $C_1 = \sum_{i=1}^{y-1} C_{1i} + \sum_{i=1}^y C_{1i}$ and $C_2 = \sum_{i=1}^{y-1} C_{2i} + \sum_{i=1}^y C_{2i}$. He sends $E(W) = (C_1, C_2)$ to Alice.
- (5) Alice decrypts (W) with the private key k to get W , where $C_1 - kC_2 = \sum_{i=1}^{y-1} M_i + \sum_{i=1}^y M_i = W$, and decodes the point W to get $w = \sum_{i=0}^{y-1} a_i + \sum_{i=0}^y a_i$. If $w = 0$, $x > y$; if $w = 1$, $x = y$; and if $w = 2$, $x < y$. Alice tells Bob the result. The protocol ends.

ALGORITHM 1: Judgement under the semihonest model.

by Alice, which is unfair to Bob. (2) In steps 3 and 4, if the ciphertext sent by Alice and Bob to each other is wrong so that neither party can get the correct result. (3) In step 5, Alice tells Bob the wrong result after decrypting, which leads to a wrong conclusion for Bob. For the above malicious attacks, a new protocol must be designed to find or render them impossible to implement. (Note: Before designing the protocol under the malicious model, we need to be clear that some malicious behaviors cannot be prevented in the ideal protocol. For example, if you enter wrong inputs in the ideal model, no matter how you detect and verify, you cannot get the correct results; similarly, if you refuse to carry out the protocol, we cannot get the results, either. Therefore, we will not consider the following behaviors when designing the protocol under the malicious model: (1) refusing to carry out the protocol, (2) inputting false data, and (3) one party terminating the protocol after obtaining the information he wants to prevent other participants from carrying out the protocol.)

To design a secure, fair, and correct protocol under the malicious model, the solution is to use cryptographic tools such as the zero-knowledge proof and cut-choose method to prevent malicious attacks that may exist in Algorithm 1. The final results are calculated by both parties at the same time.

4.2. Specific Protocol. Based on the malicious attacks that may occur in Algorithm 1 under the semihonest model, we use the above 0-1 encoding rule to design the millionaires' problem algorithm under the malicious model using the zero-knowledge proof and cut-choose method. The

framework of Algorithm 2 under the malicious model is outlined in Algorithm 3.

A specific protocol is as follows:

4.3. Correctness Analysis

- (1) The steps and positions for both Alice and Bob to execute the protocol in Algorithm 2 are identical, so we only demonstrate the possible malicious behaviors of Alice. The security analysis of the protocol is as follows:

In step (5), if Alice selects a_i that is the wrong random number, Bob happens not to choose the wrong random number a_i out of $m/2$ selected, that is, no wrong random number is detected, but in the following step (7), it happens to be selected by Bob, and Bob calculates the wrong result. The probability of success is analyzed as follows:

- ① If Alice uses the above method to commit a malicious attack, the most likely scenario for successful execution of such malicious attacks is that Alice mixes one wrong a_i in m random a_i , which maximizes the likelihood that the malicious attack will succeed. The probability of deception success in this case is $1/m$.
- ② If $m = 20$, Alice mixes one wrong a_i in m random a_i . The probability of deception success in this case is $C_{19}^{10}/C_{20}^{10} \times 1/10 = 1/200$, but if Alice mixes 10 wrong a_i in m random a_i , the probability of deception success in this case is $C_{19}^{10}/C_{20}^{10} \times 1/2 = 2.7 \times 10^{-7}$, in which case the probability of success is even smaller or negligible.

Input: Alice owns x , and Bob owns y .

Output: $P(x, y)$.

Prepare:

- (1) Alice and Bob jointly select an elliptic curve E_p and a base point G . Alice and Bob separately select their own private key k_1, k_2 ($k_1, k_2 > 0$). Then Alice and Bob calculate their public keys $K_1 = k_1G$ and $K_2 = k_2G$ and $u = aK_1$ and $v = bK_2$, respectively. Finally, Alice and Bob exchange (K_1, u) and (K_2, v) .

Alice and Bob construct their own new sets $X = (a_1, a_2, \dots, a_m)$ and $Y = (b_1, b_2, \dots, b_m)$ through x and y , where:

$$a_i = \begin{cases} 1, & i = x; \\ 0, & i \neq x. \end{cases} \quad b_i = \begin{cases} 1, & i = y; \\ 0, & i \neq y. \end{cases}$$

Start:

- (1) Alice encodes the plaintext $X = (a_1, a_2, \dots, a_m)$ onto the point M_i^a ($1 \leq i \leq m$) of the elliptic curve $E_p(a, b)$; selects m random numbers r_i^a ; encrypts each element M_i^a one by one with the public key K_1 , that is, calculates the $E(M_i^a) = (C_{1i}^a, C_{2i}^a)$, where: $C_{1i}^a = M_i^a + r_i^a K_1$ and $C_{2i}^a = r_i^a G$; obtains $E(X) = (E(M_1^a), E(M_2^a), \dots, E(M_m^a))$; and finally, sends $E(X)$ to Bob.
- (2) Bob encodes the plaintext $Y = (b_1, b_2, \dots, b_m)$ onto the point M_i^b ($1 \leq i \leq m$) of the elliptic curve $E_p(a, b)$; selects m random numbers r_i^b ; encrypts each element M_i^b one by one by using the public key K_2 , that is, calculates the $E(M_i^b) = (C_{1i}^b, C_{2i}^b)$, where: $C_{1i}^b = M_i^b + r_i^b K_2$ and $C_{2i}^b = r_i^b G$; obtains $E(Y) = (E(M_1^b), E(M_2^b), \dots, E(M_m^b))$; and finally, sends $E(Y)$ to Alice.
- (3) Alice calculates $E(Q) = (C_1, C_2)$ according to the position of data x in $E(Y)$, where $C_1 = \sum_{i=1}^{x-1} C_{1i}^b + \sum_{i=1}^x C_{1i}^b$ and $C_2 = \sum_{i=1}^{x-1} C_{2i}^b + \sum_{i=1}^x C_{2i}^b$, and sends $E(Q)$ to Bob.
Bob calculates $E(W) = (C_1', C_2')$ according to the position of data y in $E(X)$, where $C_1' = \sum_{i=1}^{y-1} C_{1i}^a + \sum_{i=1}^y C_{1i}^a$ and $C_2' = \sum_{i=1}^{y-1} C_{2i}^a + \sum_{i=1}^y C_{2i}^a$, and sends $E(W)$ to Alice.
- (4) Alice decrypts $E(W)$ using the private key k_1 , that is, calculates $C_1 - k_1 C_2 = \sum_{i=0}^{x-1} M_i^b + \sum_{i=0}^x M_i^b = W$ to obtain W point. Bob decrypts $E(Q)$ using the private key k_2 , that is, calculates $C_1' - k_2 C_2' = \sum_{i=0}^{y-1} M_i^a + \sum_{i=0}^y M_i^a = Q$ to obtain point Q .
- (5) Alice selects m random numbers d_i ($0 \leq i \leq m$) to calculate $(c_{1a}^i, c_{2a}^i) = (d_i W + K_1, W + d_i W + aG)$. Bob selects m random numbers f_i ($0 \leq i \leq m$) to calculate $(c_{1b}^i, c_{2b}^i) = (f_i Q + K_2, Q + f_i Q + bG)$. Finally, Alice and Bob exchange (c_{1a}^i, c_{2a}^i) and (c_{1b}^i, c_{2b}^i) .
- (6) With the help of the cut-choose method, Alice randomly selects the $m/2$ groups from the m groups (c_{1b}^i, c_{2b}^i) sent by Bob and publishes it and requires Bob to publish the corresponding $f_i Q$. Alice verifies: $f_i Q + K_2 = c_{1b}^i$. If the verification is passed, they terminate.
Bob randomly selects the $m/2$ groups from the m group (c_{1a}^i, c_{2a}^i) sent by Alice and publishes it and requires Alice to publish the corresponding $d_i W$. Bob verifies: $d_i W + K_1 = c_{1a}^i$. If the verification is passed, they continue the protocol or else terminate.
- (7) Alice and Bob randomly select one (c_{1b}^i, c_{2b}^i) and (c_{1a}^i, c_{2a}^i) from the remaining (c_{1b}^i, c_{2b}^i) and (c_{1a}^i, c_{2a}^i) , respectively. Meanwhile, Alice selects two random numbers h and p_1 , and Bob selects two random numbers l and p_2 . Alice calculates $c_b = h(c_{2b}^i - c_{1b}^i - W + K_2) = h(Q - W) + hlG$, $P_1 = p_1 G$, $\lambda_b = p_1 K_2$; Bob calculates $c_a = l(c_{2a}^i - c_{1a}^i - Q + K_1) = l(W - Q) + hlG$, $P_2 = p_2 G$, and $\lambda_a = p_2 K_1$. Then Alice and Bob send $c_b + P_1$ and $c_a + P_2$ to each other.
- (8) After both parties receive information from each other, Alice calculates $\omega_a = k_1(c_a + P_2)$ and $m_a = k_1 c_a$ and sends them to Bob. Bob calculates $\omega_b = k_2(c_b + P_1)$ and $m_b = k_2 c_b$ and sends them to Alice.
- (9) Alice uses the zero-knowledge proof to verify that the m_b sent by Bob is correct, that is, to prove that Bob does get the m_b by multiplying his private key k_2 with his c_b , that is, to judge whether $m_b = \omega_b - \lambda_b$ is true. Bob uses the zero-knowledge proof to verify that the m_a sent by Alice is correct, that is, to prove that Alice does get the m_a by multiplying her private key k_1 with her c_a , that is, to judge whether $m_a = \omega_a - \lambda_a$ is true. The party who fails is malicious.
- (10) Alice can get $k_2 h(Q - W)$ by calculating $m_b - hv$. If $k_2 h(Q - W) = 0$, then $Q = W$; Bob can get $k_1 l(W - Q)$ by calculating $m_a - lu$. If $k_1 l(W - Q) = 0$, then $Q = W$. If $Q = W$, it proves that the results required by both parties are correct and identical; otherwise, the protocol shall be terminated.
- (11) Finally, Alice and Bob get $\sum_{i=0}^{x-1} b_i + \sum_{i=0}^x b_i = w$ and $\sum_{i=0}^{y-1} a_i + \sum_{i=0}^y a_i = q$ by decoding points W and Q , respectively. If $q, w = 0$, then $x > y$; if $q, w = 1$, then $x = y$; and if $q, w = 2$, then $x < y$.

The protocol ends.

ALGORITHM 2: Judgement under the malicious model.

Input: x : Alice's input; y : Bob's input; G : the base point of the elliptic curve E_p ; *Co de*: encode inputs into a m degree 0-1 codes; k_1 : Alice's private key; k_2 : Bob's private key; E : encrypt. h, p_1 : Alice's random number; and l, p_2 : Bob's random number

- (1) $K_1 = k_1 G, K_2 = k_2 G$
- (2) $u = aK_1, v = bK_2$
- (3) $P_1 = p_1 G, P_2 = p_2 G$
- (4) $\lambda_b = p_1 K_2, \lambda_a = p_2 K_1$
- (5) $\text{Code}(x) = X = (a_1, a_2, \dots, a_m)$
- (6) $C_{1i}^a = M_i^a + r_i^a K_1, C_{2i}^a = r_i^a G$
- (7) $E(M_i^a) = (C_{1i}^a, C_{2i}^a)$
- (8) $E(X) = (E(M_1^a), E(M_2^a), \dots, E(M_m^a))$

ALGORITHM 3: Continued.

- (9) $Co\ de(y) = Y = (b_1, b_2, \dots, b_m)$
(10) $C_{1i}^b = M_i^b + r_i^b K_2 C_{2i}^b = r_i^b G$
(11) $E(M_i^b) = (C_{1i}^b, C_{2i}^b)$
(12) $E(Y) = (E(M_1^b), E(M_2^b), \dots, E(M_m^b))$
(13) Exchange $(K_1, E(X), u), (K_2, E(Y), v)$.
(14) $C_1 = \sum_{i=1}^{x-1} C_{1i}^b + \sum_{i=1}^x C_{1i}^b C_2 = \sum_{i=1}^{x-1} C_{2i}^b + \sum_{i=1}^x C_{2i}^b$
(15) $E(Q) = (C_1, C_2)$
(16) $C_2' = \sum_{i=1}^{y-1} C_{2i}^a + \sum_{i=1}^y C_{2i}^a$
(17) $E(W) = (C_1', C_2')$
(18) Exchange $E(Q), E(W)$
(19) $D(E(W)) = WD(E(Q)) = Q$
(20) $(c_{1a}^i, c_{2a}^i) = (d_i W + K_1, W + d_i W + aG)$ and $(c_{1b}^i, c_{2b}^i) = (f_i Q + K_2, Q + f_i Q + bG)$
(21) $((c_{1a}^i, c_{2a}^i), (c_{1b}^i, c_{2b}^i))$
(22) Alice verifies if $f_j Q + K_2 = c_{1b}^j$ and then continues or else terminates
(23) Bob verifies if $d_i W + K_1 = c_{1a}^i$ and then continues or else terminates
(24) $c_b = h(c_{2b}^j - c_{1b}^j - W + K_2) = h(Q - W) + hlG$
(25) $c_a = l(c_{2a}^i - c_{1a}^i - Q + K_1) = l(W - Q) + hlG$
(26) $m_a = k_1 c_a m_b = k_2 c_b$
(27) Exchange $(c_b + P_1, m_a), (c_a + P_2, m_b)$
(28) $\omega_a = k_1 (c_a + P_2) \omega_b = k_2 (c_b + P_1)$
(29) Exchange ω_a, ω_b
(30) $m_b = \omega_b - \lambda_b m_a = \omega_a - \lambda_a$
(31) $m_b - hv \implies k_2 h(Q - W) m_a - lu \implies k_1 l(W - Q)$
(32) if $Q = W$, then
 $D(W) = \sum_{i=0}^{x-1} b_i + \sum_{i=0}^x b_i = w$ and $D(Q) = \sum_{i=0}^x b_i + \sum_{i=0}^y a_i = q$
if $q, w = 0$, then $x > y$
else if $q, w = 1$, then $x = y$
else $x < y$
else terminate
Output: $P(x, y)$

ALGORITHM 3: Judgement under the malicious model.

③ If Alice mixes m random a_i with more than $m/2$ wrong random numbers, it will be discovered in the subsequent verification phase.

(2) In step (4), both Alice and Bob decrypt point W and point Q using their respective private keys k_1 and k_2 as follows:

$$\begin{aligned}
C_1 - k_1 C_2 &= \sum_{i=0}^{x-1} M_i^b + \sum_{i=0}^x M_i^b + K_1 \sum_{i=1}^{x-1} r_i - k_1 G \sum_{i=1}^{x-1} r_i \\
&\quad + K_1 \sum_{i=1}^x r_i - k_1 G \sum_{i=1}^x r_i = \sum_{i=0}^{x-1} M_i^b + \sum_{i=0}^x M_i^b = W, \\
C_1' - k_2 C_2' &= \sum_{i=0}^{y-1} M_i^a + \sum_{i=0}^y M_i^a + K_2 \sum_{i=1}^{y-1} r_i - k_2 G \sum_{i=1}^{y-1} r_i \\
&\quad + K_2 \sum_{i=1}^y r_i - k_2 G \sum_{i=1}^y r_i = \sum_{i=0}^{y-1} M_i^a + \sum_{i=0}^y M_i^a = Q.
\end{aligned} \tag{5}$$

(3) The (c_{1a}^i, c_{2a}^i) and (c_{1b}^i, c_{2b}^i) published in step (5) do not leak any information because their own random numbers are added.

(4) In step (7), Alice and Bob calculate as follows:

$$\begin{aligned}
c_b &= h(c_{2b}^j - c_{1b}^j - W + K_2) \\
&= h(Q + f_j Q + bG - f_j Q - K_2 - W + K_2) \\
&= h(Q - W) + hlG, \\
c_a &= l(c_{2a}^i - c_{1a}^i - Q + K_1) \\
&= l(W + d_i W + aG - d_i W - K_1 - Q + K_1) \\
&= l(W - Q) + hlG.
\end{aligned} \tag{6}$$

Alice and Bob then send $c_b + P_1$ and $c_a + P_2$ to each other.

(5) In step (10), Alice and Bob get the right results.

Alice uses the zero-knowledge to prove that the m_b sent by Bob is correct; the result obtained by calculating $m_b - hv$ is correct, that is,

$$\begin{aligned}
m_b - hv &= m_b - hlK_2 \\
&= k_2 c_b - hlk_2 G \\
&= k_2 h(Q - W) + k_2 hlG - hlk_2 G \\
&= k_2 h(Q - W).
\end{aligned} \tag{7}$$

After Bob uses zero-knowledge to prove that the m_a sent by Alice is correct; the result obtained by calculating $m_a - bu$ is correct, that is:

$$\begin{aligned}
m_a - lu &= m_a - hlK_1 \\
&= k_1c_a - hlk_1G \\
&= k_1l(W - Q) + k_1hlG - hlk_1G \\
&= k_1l(W - Q).
\end{aligned} \tag{8}$$

- (6) In step (11), Alice and Bob decode point $W = \sum_{i=0}^{x-1} M_i^b + \sum_{i=0}^x M_i^b$ and point $Q = \sum_{i=0}^{y-1} M_i^a + \sum_{i=0}^y M_i^a$ to get $\sum_{i=0}^{x-1} b_i + \sum_{i=0}^x b_i = w$ and $\sum_{i=0}^{y-1} a_i + \sum_{i=0}^y a_i = q$, respectively.
- (7) The encoding of plaintexts on points W and Q in steps (1) and (2) and the decoding of points W and Q in step (11) can be referred to reference [23].
- (8) The whole process does not leak any confidential information, and both parties can obtain results independently, avoiding unfairness caused by one party telling the other party the result.
- (9) In many cases, the data range is known to all parties in reality. For example, if two students want to compare their grades, then the data range is $(a_1, a_2, \dots, a_{100})$, that is known to all parties; if two companies at the same level want to compare their assets, the data range may be $(a_{1M}, a_{2M}, \dots, a_{100M})$, but a company's assets are often sparsely rather than densely distributed on the data range. Assets can only be a few scales, and the data range is very small. Therefore, they know the data range. The data range does not leak any information about its private data. Generally speaking, all the numbers compared in SMC are comparable. If these figures are comparable, both parties will know their scope. Ordinary companies will never compare their assets with Microsoft because they are not comparable. However, we have to say that although the data range is known, if the data range is large, the computational complexity of the protocol will be very high, so the protocol becomes impractical.

4.4. Security Proof. Algorithm 2 under the malicious model is proved as follows.

Definition 2. Algorithm 2 is secure under the malicious model.

Proof. This proving process borrows a trusted third party (TTP). We set the actual policy pair as $\bar{A} = (A_1, A_2)$, the ideal policy pair as $\bar{B} = (B_1, B_2)$, F as the output, and S as the message sequence received by A_2 in the zero-knowledge proof process. We want to prove that the security of the protocol under the malicious model is to prove that when Algorithm 2 is executed, the implementation of malicious behaviors in the actual protocol calculation will not affect the correct output, that is:

$$\{\text{REAL}_{\bar{A}}(W, Q)\} = \{\text{IDEAL}_{\bar{B}}(W, Q)\}. \tag{9}$$

In Algorithm 2, the malicious behaviors are not allowed for both Alice and Bob at the same time, so there are two

scenarios: Alice or Bob is honest. Here, A_1, B_1 and A_2, B_2 represent Alice and Bob, respectively.

- (1) If A_1 is honest and A_2 is dishonest, then:

$$\text{REAL}_{\bar{A}}(W, Q) = \{F(W, A_2(Q)), A_2((c_{1a}^i, c_{2a}^i), m_a, S)\}. \tag{10}$$

- ① Since A_1 is honest, B_1 sends a correct W to TTP, and the protocol will be executed correctly.
- ② What B_2 sends to TTP depends on the actual selection of A_2 . B_2 sends Q to A_2 under the ideal model. A_2 sends $A_2(Q)$ to B_2 in the practical cases, and B_2 sends $A_2(Q)$ to TTP. Finally, TTP outputs $F(W, A_2(Q))$.
- ③ Ideally, B_2 uses the $F(W, A_2(Q))$ sent by TTP to try to get $\text{view}_{B_2} F(W, A_2(Q))$ that is indistinguishable from the $\text{view}_{A_2} F(W, A_2(Q))$ calculated by A_2 in practice and make it the output of A_2 in the practical cases.

That is, B_2 selects W' to make $F(A_1(W), Q) = F(A_1(W'), Q)$, performs all the calculations in Algorithm 2, obtains m'_a and c_{1a}^i, c_{2a}^i , and records the received sequence S' in the zero-knowledge proof. Thus, the protocol proceeds, and we get

$$\{\text{IDEAL}_{\bar{B}}(W, Q)\} = \{F(W, A_2(Q)), A_2((c_{1a}^i, c_{2a}^i), m'_a, S')\}. \tag{11}$$

Because ciphertexts are encrypted ideally and practically using the same probability algorithm, there are $c_{1a}^i c \equiv c_{1a}^i$ and $c_{2a}^i c \equiv c_{2a}^i$. The random number a_i is indistinguishable from a'_i , so $\{\text{REAL}_{\bar{A}}(W, Q)\} = \{\text{IDEAL}_{\bar{B}}(W, Q)\}$.

- (2) If A_1 is dishonest and A_2 is honest, there are two situations:

Actually, A_1 completes the zero-knowledge proof and publishes the results:

$$\text{REAL}_{\bar{A}}(W, Q) = \{A_1((c_{1b}^i, c_{2b}^i), m_b, S), F(W, Q)\}. \tag{12}$$

Actually, A_1 does not publish the results or execute the zero-knowledge proof:

$$\text{REAL}_{\bar{A}}(W, Q) = \{A_1((c_{1b}^i, c_{2b}^i), m_b, S), \perp\}. \tag{13}$$

- ① Because A_2 is honest, B_2 will send correct Q to TTP, and the protocol will be carried out correctly.
- ② What B_1 will send to TTP depends on the choice of A_1 in the practical situation. Ideally, B_1 sends W to A_1 ; practically, A_1 sends $A_1(W)$ to B_1 , and B_1 sends $A_1(W)$ to TTP. Finally, TTP outputs $F(A_1(W), Q)$.
- ③ If A_1 does not publish the results or do not conduct the zero-knowledge proof in practice, B_2 in the ideal model will get \perp from TTP.
- ④ Ideally, B_1 uses the $F(A_1(W), Q)$ sent by TTP to try to obtain $\text{view}_{B_1}(A_1(W), Q)$ that is indistinguishable

from $\text{view}_{A_1}(A_1(W), Q)$ calculated by A_1 in the practical situation and to make it to be the output of A_1 in the practical situation.

That is, B_1 selects Q' to make $F(A_1(W), Q') = F(A_1(W), Q)$. And Algorithm 2 is carried out. Finally, under the ideal model, B_1 obtains m'_b and c'_{1b}, c'_{2b} and records the sequence S' received through the zero-knowledge proof. In this way, we get the following.

Ideally, when B_1 does not publish results to B_2 via TTP:

$$\text{IDEAL}_{\bar{B}}(W, Q) = \left\{ A_1 \left(\left(c'_{1b}, c'_{2b} \right), m'_b, S' \right), \perp \right\}. \quad (14)$$

Ideally, when B_1 announces results to B_2 via TTP:

$$\text{IDEAL}_{\bar{B}}(W, Q) = \left\{ A_1 \left(\left(c'_{1b}, c'_{2b} \right), m'_b, S' \right), F(A_1(W), Q) \right\}, \quad (15)$$

where c'_{1b}, c'_{2b} , and c^i_{1b}, c^i_{2b} are encrypted by the same ECC encryption algorithm. m'_b and m_b are computed by both random numbers and constant operations. The zero-knowledge proof guarantees $S'^c \equiv S$. Therefore, for any algorithm in the practical protocol $\bar{A} = (A_1, A_2)$, there exists $\bar{B} = (B_1, B_2)$ in the ideal protocol, which makes

$$\{\text{IDEAL}_{\bar{B}}(W, Q)\} \stackrel{\epsilon}{\equiv} \{\text{REAL}_{\bar{A}}(W, Q)\}. \quad (16)$$

Thus, the protocol's security is proven.

5. Efficiency Analysis

5.1. Computational Complexity. Reference [24] proposed a protocol to solve the millionaires' problem based on the decision Diffie-Hellman hypothesis (DDH) and performed $4nt + n$ modular multiplications. Reference [25] designed a millionaires' problem comparison protocol with 0-1 coding rules based on ElGamal encryption, which needs $(2m + 3)\log P + 5m$ modular multiplications. Reference [22] designed an antimalicious millionaires' problem protocol based on the Paillier encryption and performed $10m \log N + 2$ modular multiplications.

During the execution of Algorithm 1, the computational complexity mainly includes: m times ECC encryption operations of Alice and 1 time ECC decryption operation of Bob, with a total of $2m + 1$ modular multiplications. During the execution of Algorithm 2, the computational complexity mainly includes: m times ECC encryption operations and 1 time ECC decryption operation of Alice and several modular multiplication operations during message verification. A total of $11m + 10$ modular multiplications are performed, and the rest are ordinary multiplication and addition operations, which can be ignored.

5.2. Communication Complexity. There are two rounds of communication in Algorithm 1; reference [24] carried out three rounds of communication; reference [25] carried out three rounds of communication; reference [22] carried out three rounds of communication; and Algorithm 2 carried out six rounds of communication, as shown in Table 2.

5.3. Experimental Simulation. To verify the validity of the above protocols more intuitively, we compare Algorithm 2 with reference [22, 24, 25]. The experimental environment is Windows10 (64 bit) operating system, Intel (R) Core (TM) i7-5500U CPU @ 2.40 GHz processor, 8.00 GB memory, and the experiment is carried out by Python language.

The Paillier encryption, ElGamal encryption, and GM encryption have the same size of inputs in the experiment, and the time of protocol preprocessing is ignored in the experiment. Figure 2 is a comparison of the protocol time consumption in experiment 1 with the increase of modulus. The data held by each participant in experiment 1 is an integer from 0 to 100 (set length is set to 100). The average execution time (the ordinate coordinate) of the four protocols is calculated under 128, 256, 512, and 1,024 bit modules (the horizontal coordinate). As can be seen from Figure 2, the time consumed by Algorithm 2 is lower than those of other references.

Figure 3 is a comparison of the execution time of experiment 2 with the increase of data range. In experiment 2, each participant held the input data in the range of 0-100, 100-200, 200-300, 300-400, 400-500, 500-600, 600-700, 700-800, 800-900, and 900-1,000 under the same module. As can be seen from Figure 3, Algorithm 2 takes less time in different data ranges than other protocols, and the time consumption is relatively stable.

The results show that under the same security performance, Algorithm 2 is slightly more efficient than reference [24] and has obvious advantages than Reference [22, 25]. Algorithm 2 can resist malicious attacks and has higher security and greater practical value. (Note: For Algorithm 2, increased the bitcoin commitment, cut-choose, and zero-knowledge proof methods will result in significantly higher computational complexity and lower execution efficiency, making malicious model protocols no more efficient than semihonest model protocols. However, preprocessing or computing outsourcing can be used to improve efficiency, and both methods are available in Algorithm 2).

5.4. Applications. This paper uses efficient ECC encryption to design and study the classic millionaires' problem in SMC. It not only solves the problem of comparison between two numbers but also distinguishes whether two numbers are equal or not. The protocol can be widely applied to sort confidentially. Alice has $X = (a_1, a_2, \dots, a_m)$; Bob has y ; Bob wants to query the ranking position of y in X ; and both sides do not want to expose any information about X and y . This problem is an important application of SMC in data query and has wide application prospects, such as secret ranking of college entrance examination results: after the college entrance examination, candidates want to check their ranking in the reported candidates, and the school does not want to disclose any information about other candidates. The problem can be solved by our protocol.

The same method can be applied to the smart contract quality evaluation in the blockchain. Blockchain technology is considered to be the next generation of disruptive core technology after steam engine, power, and Internet. In order

TABLE 2: Performance comparison of the efficiency of the four protocols in terms of computational complexity, number of communication rounds, and whether they can resist malicious attacks.

Protocol	Fair for both parties	Computational complexity (modular multiplications)	Communication (rounds)	Resist malicious attacks
Algorithm 1	No	$2m + 1$	2	×
Reference [24]	No	$4nt + n$	$3n \log P$	×
Reference [25]	No	$(2m + 3)\log P + 5m$	3	×
Reference [22]	Yes	$10m \log N + 2$	3	✓
Algorithm 2	Yes	$11m + 10$	6	✓

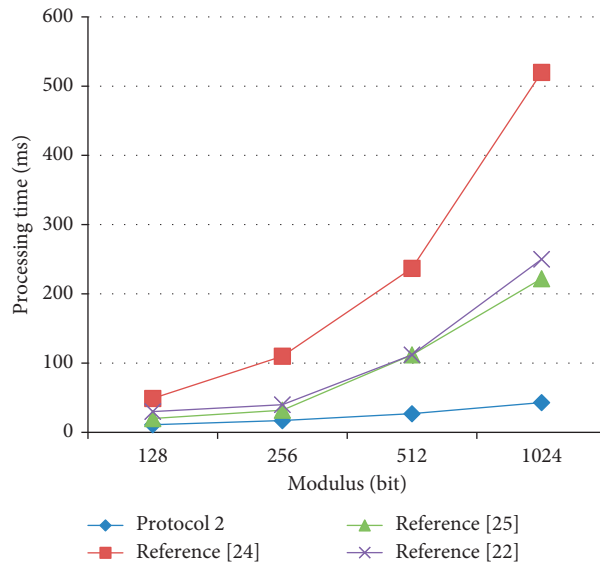


FIGURE 2: Time consumption comparison of different modules. The data held by each participant is an integer from 0 to 100 (set length is set to 100).

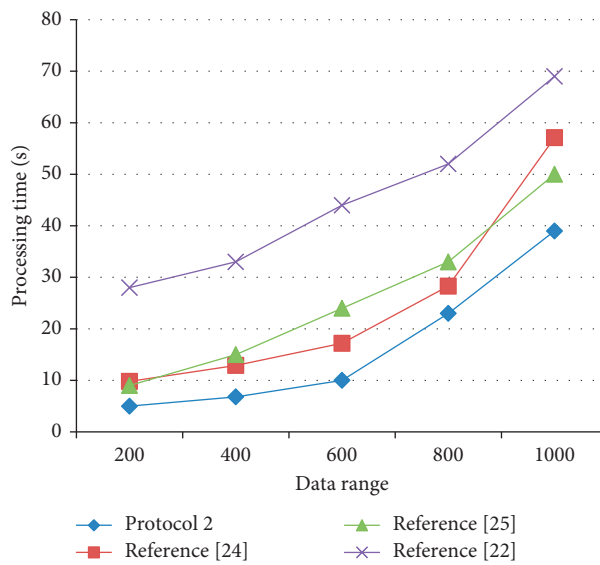


FIGURE 3: Time consumption comparison for different input ranges under the same module.

to establish a high-quality blockchain application environment, excellent smart contract developers will be rewarded by obtaining some form, such as tokens. However, there is no good evaluation method for the quality of the smart contract. Most of them are sorted by the number of contract calls and the total amount of contract transactions. Therefore, using our protocol to construct the sorting method, the screened high-ranking users have strong authenticity and security and are not easy to forge.

6. Summary and Prospect

As the cornerstone of SMC, the millionaires' problem is still under constant researches by experts and scholars. However, most of the schemes are designed under the semihonest model, which cannot resist malicious attacks and affect the practical application of the protocols. This paper uses the 0-1 encoding method and ECC encryption algorithm, first designs a semihonest model protocol, and then improves it for malicious behaviors. The millionaires' problem protocol under the malicious model solves the problem of malicious attacks in practical applications. By comparing with the existing protocols, our protocol is more efficient and practical.

Data Availability

The algorithm 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 study was supported by the fund project, National Natural Science Foundation of China (92046001 and 61962046); Inner Mongolia Natural Science Foundation (2021MS06006); Baotou Kundulun District Science and Technology Planning Project (YF2020013); Inner Mongolia Discipline Inspection And Supervision Big Data Laboratory open project fund (IMDBD2020020); Major Science And Technology Projects in Inner Mongolia (2019ZD025); basic scientific research business fee project of Beijing Municipal Commission of education (110052972027); the Scientific Research Launch Funds of North China University of Technology (110051360002); and Beijing Urban Governance Research Base of North China University of Technology.

References

- [1] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pp. 160–164, IEEE press, Chicago, IL, USA, June 1982.
- [2] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, ACM*, pp. 218–229, Montreal, Canada, June 1987.
- [3] O. Goldreich, *Foundations of Cryptography-Volume 2: Basic Applications*, Cambridge University Press, Cambridge, UK, 2009.
- [4] T. Li, Z. Wang, G. Yang, and Y. Cui, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, 2021.
- [5] Y. Chen, J. Sun, Y. Yang, and T. Li, "PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, vol. 37, 2021.
- [6] T. Wang, W. Ma, and W. Luo, "Information sharing and secure multi-party computing model based on blockchain," *Computer science*, vol. 46, no. 9, pp. 162–168, 2019.
- [7] J. Liu, Y. Tian, Y. Zhou, Y. Xiao, and N. Ansari, "Privacy preserving distributed data mining based on secure multi-party computation," *Computer Communications*, vol. 153, pp. 208–216, 2020.
- [8] G. Xu, Y. Cao, S. Xu et al., "A novel post-quantum blind signature for log system in blockchain," *Computer Systems Science and Engineering*, vol. 41, no. 3, pp. 945–958, 2022.
- [9] M. Blatt, A. Gusev, Y. Polyakov, and S. Goldwasser, "Secure large-scale genome-wide association studies using homomorphic encryption," *Proceedings of the National Academy of Sciences*, vol. 117, no. 21, pp. 11608–11613, 2020.
- [10] X. Liu, R. Zhang, G. Xu, X.-B. Chen, and N. N. Xiong, "Confidentially judging the relationship between an integer and an interval against malicious adversaries and its applications," *Computer Communications*, vol. 180, pp. 115–125, 2021.
- [11] L. Feng, Y. Jie, and L. Zhibin, "A secure multi-party computation protocol for universal data privacy protection based on blockchain," *Journal of Computer Research and Development*, vol. 58, no. 2, p. 281, 2021.
- [12] D. Li, X. Liao, T. Xiang, J. Wu, and J. Le, "Privacy-preserving self-serviced medical diagnosis scheme based on secure multi-party computation," *Computers & Security*, vol. 90, Article ID 101701, 2020.
- [13] S. Parthasarathy, A. Harikrishnan, and G. Narayanan, "Secure distributed medical record storage using blockchain and emergency sharing using multi-party computation," in *Proceedings of the 2021 11th IFIP International Conference On New Technologies, Mobility And Security (NTMS)*, pp. 1–5, Paris, France, April 2021.
- [14] W. Liu, Y.-B. Wang, A.-N. Sui, and M.-Y. Ma, "Quantum protocol for millionaire problem," *International Journal of Theoretical Physics*, vol. 58, no. 7, pp. 2106–2114, 2019.
- [15] M. Hezaveh and C. Adams, "An efficient solution to the socialist millionaires' problem," in *Proceedings of the 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1–4, IEEE, Windsor, Canada, April 2017.
- [16] S. Li and M. Zhang, "Efficient solutions to the problem of blind millionaires," *Journal of Computer Science*, vol. 43, no. 9, pp. 1755–1768, 2020.
- [17] J. Zhang, H. Zheng, B. Ge, Y. Tang, and Q. Ye, "An efficient millionaire problem protocol and its application," *Computer Engineering*, vol. 47, no. 2, pp. 168–175, 2021.
- [18] S. Ibrahim and A. Alharbi, "Efficient image encryption scheme using henon map, dynamic S-boxes and elliptic curve cryptography," *IEEE Access*, vol. 8, pp. 194289–194302, 2020.
- [19] F. De Rango, G. Potrinio, M. Tropea, and P. Fazio, "Energy-aware dynamic internet of things security system based on elliptic curve cryptography and message queue telemetry

- transport protocol for mitigating replay attacks,” *Pervasive and Mobile Computing*, vol. 61, Article ID 101105, 2020.
- [20] R. Qazi, K. N. Qureshi, F. Bashir, N. U. Islam, S. Iqbal, and A. Arshad, “Security protocol using elliptic curve cryptography algorithm for wireless sensor networks,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 547–566, 2021.
- [21] A. De Santis, S. Micali, and G. Persiano, “Non-interactive zero-knowledge proof systems,” in *Proceedings of the Conference on the Theory and Application of Cryptographic Techniques*, pp. 52–72, Springer, Santa Barbara, CA, USA, August 1987.
- [22] S. Li, W. Wang, and R. Du, “Solutions to the millionaire problem against malicious enemies,” *Chinese Science: Information Science*, vol. 51, no. 1, pp. 75–88, 2021.
- [23] B. Yang, *Modern Cryptography* pp. 124–128, Tsinghua University Press, Beijing, China, 4th edition, 2017.
- [24] M. Liu, Y. Luo, P. Nanda, S. Yu, and J. Zhang, “Efficient solution to the millionaires’,” *Computational Intelligence*, vol. 35, no. 3, pp. 555–576, 2019.
- [25] Z. Li, L. Chen, Z. Chen, Y. Liu, and T. Gao, “An efficient millionaire problem protocol based on 1-r coding and its application,” *Acta cryptographica Sinica*, vol. 6, no. 1, pp. 50–60, 2019.

Research Article

A Privacy Protection Scheme for Facial Recognition and Resolution Based on Edge Computing

Junhua wu ¹, Wenzhen Feng ¹, Guopeng Liang ¹, Tiantian Wang ¹, Guangshun Li ¹,
and Yuanwang Zheng²

¹School of Computer Science, Qufu Normal University, Rizhao 276826, China

²Shandong Huatong Used Car Information Technology Limited Company, Jining 272600, China

Correspondence should be addressed to Guangshun Li; guangshunli@qfnu.edu.cn

Received 25 November 2021; Accepted 25 January 2022; Published 10 March 2022

Academic Editor: Yuling Chen

Copyright © 2022 Junhua wu 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.

Facial recognition and resolution technology have extensive application scenarios in the era of big data. It ensures the consistency of personal identity in physical space and cyberspace by establishing correspondence between physical objects and network entities. However, massive data brings huge processing pressure to cloud service, and there are data leakage risks about personal information. To address this problem, we propose a privacy security protection scheme for facial recognition and resolution based on edge computing. Firstly, a facial recognition and resolution framework based on edge computing is established, which improves the communication and storage efficiency through task partition and relieves the pressure of cloud computing. Then, a verifiable deletion scheme based on Hidden CP-ABE is proposed to provide fine-grained access control and ensure the safe deletion of target data in the cloud. Moreover, after applying the verifiable deletion method, the safe deletion of the target data in the cloud can be achieved. Finally, the simulation results show the effectiveness and security of the proposed scheme.

1. Introduction

Internet of Things (IoT) devices are very common in our daily life, and more and more physical entities are connected to cyberspace. A series of intelligent applications derived from IoT such as smart home, smart medical, and smart grid are profoundly changing social public services and people's daily life [1]. As a key biometric technology to authenticate personal identity, facial recognition and resolution technology can collect any object that needs monitoring and interaction through information sensors. It has been widely used in security fields such as smart card user authentication, criminal investigation, and access control management [2]. Because facial image involves a large amount of private information including identity identifiers, and its recognition and resolution process requires high computing, high storage, and high communication capabilities. This information is usually uploaded to the cloud server in an unencrypted state. Once the cloud server is attacked, privacy leakage is likely to occur [3].

Some researchers have adopted cloud computing to improve computing, storage, and communication capabilities. But limited network bandwidth has become the bottleneck of this centralized processing architecture. As a distributed computing paradigm, edge computing can provide an intermediate layer between the cloud and terminal devices and use the computing power of edge devices to provide efficient services [4]. Compared with the traditional cloud server model, we introduce edge computing technology into the facial recognition and resolution framework, forming a three-tier distributed architecture composed of cloud, edge, and terminal devices. The cloud server offloads some computing tasks to the edge server and performs some operations on it, in order to reduce the interaction with the cloud server. Therefore, we propose a face recognition and resolution framework based on edge computing, which can not only ease the tension between computing-intensive applications and resource-constrained mobile devices, but also reduce the long delay caused by the

interaction between devices and the cloud to improve data processing efficiency [5].

The cloud server provides finite privacy protection under the complex network environment, and the data stored in them is vulnerable to malicious attacks [6]. At present, the shared files on most data storage servers are generally expressed in plain text or simply encrypted. This traditional security mechanism consumes more resources and bandwidth. In CP-ABE encryption, the ciphertext is associated with the access structure, and the key is associated with the attribute set. Only when the decryptor attribute meets the ciphertext-related access policy can the ciphertext be decrypted. However, any user who can obtain the ciphertext can see the content of the access structure, so it may cause information leakage of the decryption party or the encryption party. When data in the cloud is no longer available, the lack of effective deletion will lead to serious problems such as abuse and theft [7]. Physical destruction is obviously the most effective way to delete stored data, but we need to delete it at the file level, and other data can still remain. Therefore, the privacy and security issue of data transmission and data storage in facial recognition and resolution needs to be resolved urgently.

This paper aims to optimize the security and privacy protection scheme of facial recognition and resolution framework and further ensure data security during transmission and storage. We introduce the edge nodes to relieve the bandwidth pressure of transmission and improve the efficiency of calculation. And we apply the Hidden CP-ABE scheme to the data before it is transmitted to the cloud. Furthermore, we adopt a verifiable deletion scheme to ensure the “true deletion” of cloud data.

The main contributions can be summarized as follows:

- (1) We establish a face recognition and resolution framework based on edge computing, and it can reduce the network bandwidth pressure of the cloud server through completing the recognition and resolution of facial images on the edge server.
- (2) We propose a verifiable deletion scheme based on the Hidden CP-ABE, which encrypts the data before uploading it to the cloud. According to the requirements of the data owner, a verifiable deletion method is adopted to confirm the deletion of the target data in the cloud, to prevent attackers from accessing the relevant data after “false deletion.” Consequently, the data storage security is ensured in the cloud.
- (3) Experimental results show that the transmission bits are effectively reduced under this scheme, and the facial recognition and resolution framework can provide more secure and efficient services.

The rest of the paper is organized as follows. Section 2 briefly introduces the relevant work. Section 3 introduces the system model. Section 4 introduces the verifiable deletion scheme of cloud data in facial recognition and resolution. Section 5 carries on the simulation experiment to test the validity of our scheme. Section 6 draws the conclusion.

2. Related Work

The privacy security of facial recognition and resolution is of great significance to ensure the security of the IoT. Researchers at home and abroad have conducted some studies on the architecture and security of the IoT.

Compared with the traditional cloud computing network, the layered distributed computing architecture based on edge computing and cloud computing can solve the problems of data transmission efficiency and network bandwidth more effectively [8]. In order to meet the requirements of high computing power and reduce the corresponding costs, most operators outsource huge amounts of data and computing tasks to cloud servers [9]. However, the cloud server is generally relative far away from the position of the service request, which may lead to a longer delay and lower user satisfaction, especially, applications of face recognition that require swift feedback [10]. Therefore, in order to provide highly responsive cloud services. In [11], Shi et al. proposed to sink the computing and storage center to the edge of the Internet near the image acquisition equipment to reduce the communication delay. In [12], Ning et al. proposed a new information retrieval scheme that better reduces the computing burden and network transmission load of the cloud by introducing edge computing technology. In [13], Yu et al. improved the Label Distribution Protocol (LDP) algorithm and the centralized algorithm for global interference location. And they designed a distributed Centralized and Localized Traversal (CLT) algorithm on this basis, which only lost a constant part of the optimal scheduling and significantly reduced the time complexity of the algorithm and transmission delay. In [14], Barbieri et al. proposed an independent health management architecture, which executes efficient and fast algorithms on edge servers and combines them with other algorithms on cloud servers, showing a certain degree of robustness.

All of the above schemes have optimized the delay problem, but the edge computing architecture determines that it faces new security and privacy challenges [15]. Especially in the field of biological information recognition, the problem of information transmission security between edge server and cloud server has not been effectively solved. After many sensors collect our facial data, if the data is sent to the cloud server without encryption, there will be the risk of privacy leakage of eavesdropping or tampering, reducing the reliability of network transmission [16, 17]. Regarding the problems above, researchers put forward many privacy protection methods for biometric recognition.

In [18], Ma et al. proposed a lightweight adaptive enhanced facial recognition framework based on additive secret sharing and edge computing, and designed a series of interactive protocols for privacy protection integrated classification. It not only improves the fault tolerance rate of the system, but also makes it possible to calculate encrypted facial features between the two deployed edge servers. To further ensure security and prevent malicious client attacks, in [19], Im et al. proposed a smart phone face authentication system. The face feature vector is stored on the cloud server in encrypted form, and the Euclidean distance matching

score is calculated by using homomorphic encryption, which makes faster verification speed and higher verification rate. However, the related calculations in traditional encryption schemes are usually carried out independently in the cloud. In [20], in order to improve the efficiency of ciphertext expansion, Wang et al. proposed a dynamic multikey scheme, which hides the key by public key and uniform random matrix to make the parties jointly deliver and reduce the workload of the cloud.

CP-ABE algorithm is a promising solution for fine-grained access control. In order to implement more fine-grained access control over transmitted data, in [21], Qi et al. designed an industrial data access control scheme that outsources tasks to cloud services, providing stronger security guarantees. More importantly, the scheme carries out item-level data protection to prevent key disclosure. However, the system overhead of the above scheme is relatively high, and the traditional CP-ABE scheme may leak sensitive information embedded in the ciphertext access structure. Therefore, in [22], Zhang et al. proposed a fixed-length ciphertext distributed CP-ABE scheme that completely hides the access policy. In [23], Tian et al. proposed a lightweight completely hidden protection access control scheme based on attributes, which achieves complete privacy protection through three key stages of key generation, access control, and partial decryption. In [24], Yu et al. proposed an intelligent IoT privacy protection scheme based on multi-permission CP-ABE to prevent the platform from prying on user data. The above three optimized CP-ABE schemes effectively hide the attribute values and reduce the communication overhead and computing overhead of the client to a certain extent.

In the above work, researchers have proposed a number of schemes for encrypting data before it is uploaded. However, the common challenge faced is the security of the “to be deleted” data stored in the cloud. All the stored data can be deleted at one time by physically damaging the hardware, but the purpose of deleting is to prevent attackers from continuing to access the data after the deletion, so this method is effective but not advisable [25]. To better comply with the processing rules of user private data, we need to delete the target data from the storage media at the file level, making it unrecoverable and leaving the rest of the data unaffected [26]. Therefore, some researchers consider to solve such problems by fine-grained attribute revocation, but it usually results in excessive key management overhead and high computational costs.

For this reason, in [27], Yeh et al. proposed a cloud-based fine-grained health information access control framework. It has the functions of dynamic data audit and attribute revocation. In [28], Miao et al. used a hierarchical commitment method for updating. This structure can simultaneously satisfy the private verifiability and public verifiability, but when the client continuously inserts new data in the same index of the database, the hierarchical commitment level has increased linearly. In this case, the load of cloud computing and storage will increase accordingly. Therefore, in [29], Ma et al. proposed a fine-grained access control mechanism that can be

undone by the storage, computing, and management capabilities of the cloud to achieve efficient user's revocation based on fine-grained attributes. In [30], Edemac et al. proposed an expressible and collusion-resistant new access control scheme, which further realized forward and backward security. But how to eliminate the dependence on a single trusted authority remains to be resolved. In [31], Yu et al. proposed a new blockchain-based IoT system to solve the immutability of traditional blockchain and the incompatibility of attribute updates. In summary, data security protection is a quite challenging problem. In the field of facial recognition and resolution, how to ensure the security of data transmission and data storage at the same time is worth further study.

3. Network Model and definitions

In this section, we first introduce the facial recognition and resolution framework based on edge computing. Then, we analyze the functions of each module and the specific process of facial recognition and resolution. In addition, we analyze the potential risks in the transmission and stored process based on this framework.

3.1. Facial Recognition and Resolution Framework Based on Edge Computing. Compared with the traditional cloud computing model, in order to reduce the throughput of the transmission channel and the computing load of the cloud, a facial recognition and resolution framework based on edge computing is proposed.

The framework consists of three main parts: client, edge server, and cloud server. The client usually consists of terminal devices such as mobile phones and computers with cameras. The edge server includes two kinds of resolution servers: one is an image parsing server, and the other is an information parsing server. The cloud servers usually consist of a management server and a data center in the cloud. Figure 1 shows the facial recognition and resolution framework. The functions of each functional module are as follows.

Client: It is responsible for temporarily storing the original facial images collected by the visual detection equipment, and initiating facial recognition and resolution services to the edge server. After the identity is successfully matched, it returns the resolved identity information to the client.

Edge Server: It mainly includes an image resolution server and information resolution server. Resolution is performed closer to the user side without delivering the information to the cloud. The image resolution server is used to resolve the original facial image into the corresponding facial identifier. The information resolution server is used to resolve each piece of personal information registered by the client into a corresponding serial number, bind the facial identifier to each serial number, and then send it to the cloud for identity matching.

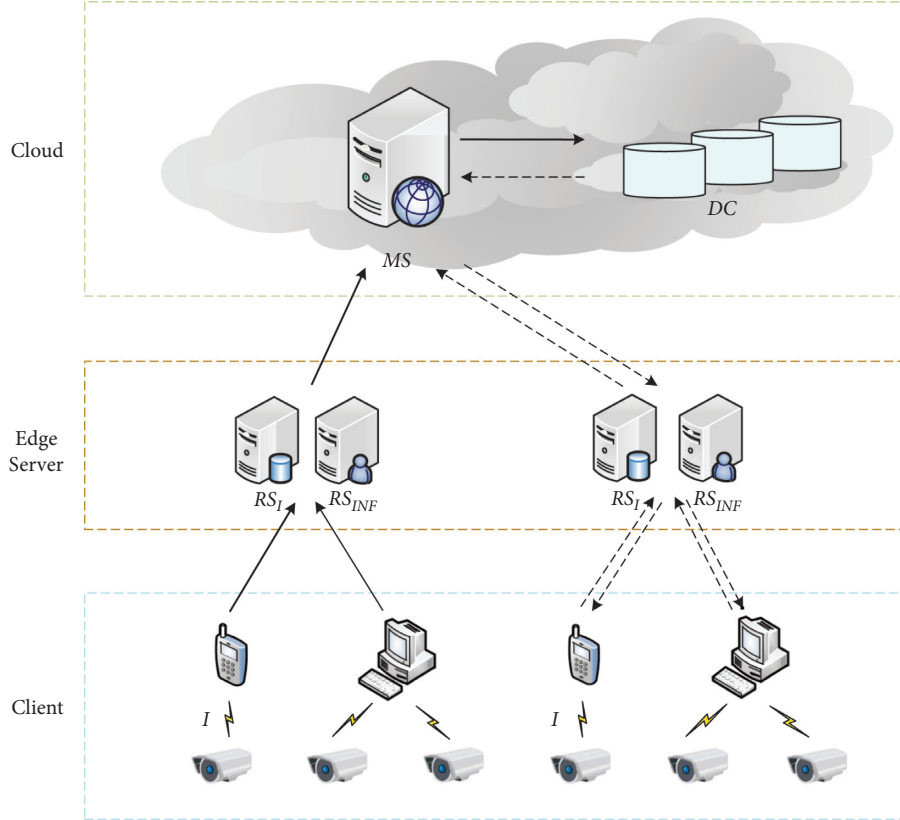


FIGURE 1: Framework for facial recognition and resolution.

Data Center (DC): It is responsible for storing the information from the edge server and performing preliminary matching operations in the existing database.

Management Server (MS): It is responsible for receiving the information from the edge server, and properly scheduling and distributing it to the data center for matching operations during facial resolution.

3.2. Facial Recognition

- (1) The client device collects the original facial image and uploads it to the client.
- (2) The client receives the original facial image and registers the necessary personal information. It initiates a facial recognition service to the edge server and sends the obtained original facial image and personal information to the edge server after establishing a network connection.
- (3) The edge server receives the original facial image and personal information. The image resolution server resolves the original facial image and generates the facial identifier. Generate facial identifiers by performing algorithms such as face detection and preprocessing, feature extraction, and facial identifier generation. At the same time,

TABLE 1: Symbol definition list of key negotiation process.

Term	Description	Page
q	A prime number	6
g	Generator	6
ID_{RS}, ID_{MS}	The identifier of RS, the identifier of MS	6
x_{RS}, x_{MS}	The private key of RS, The private key of MS	6
y_{RS}, y_{MS}	The public key of RS, The public key of MS	6
K_{RS}, K_{MS}	The secret key of RS, the secret key of MS	7
$K_{RS,MS}$	The shared secret key of RS and MS	7

the personal information is parsed into the corresponding serial number by the information resolution server. The facial identifier and serial number are bound and uploaded to the cloud together.

- (4) The management server in the cloud receives the facial identifier and serial number and then stores them in the data center. Finally, the successful registration flag is returned to the edge server, and then, returned to the client.

At this point, the whole facial recognition process is completed. It successfully realized the identity information registration and transform individual faces in physical space into identity identifiers in information space.

3.3. Facial Resolution

- (1) The client device obtains the original facial image of the person being tested, sends a facial resolution request to the edge server, and then sends the original facial image to the edge server.
- (2) Similar to the facial recognition process, the image resolution server in edge sever generates facial identifier from the received original facial image by performing algorithms such as face detection, pre-processing, feature extraction, and facial identifier generation. Edge server initiates a facial resolution service to the cloud and sends the facial identifier to the cloud after establishing network connection.
- (3) The cloud receives the facial identifier. Firstly, the management server in the cloud initially matches the facial identifier with the existing facial identifier in the data center. After the match is successful, the corresponding serial number of the facial identifier is returned to the management server, which in turn is returned to the edge server. The information resolution server in the edge server reversely parses the returned serial number into the corresponding personal information. Finally, the edge server returns the obtained personal information to the client and displays it to the terminal user.

At this point, the whole facial resolution process is completed, realizing the matching of the facial image in the physical space and the personal identity in cyberspace, and ensuring the consistency of the corresponding relationship between them.

3.4. Security Issues. Due to the openness of channels and the sensitivity of data, with the maximization of business purposes, both edge and cloud computing sectors have a strong interest in user information. There are many potential security threats in practical application, so it should be guaranteed from the following three aspects.

- (1) Security of data transmission process. The process of facial recognition and resolution involves the transmission from the edge server to the cloud server, which is extremely vulnerable to malicious attacks. Facial information is usually closely associated with sensitive personal information such as health care or financial records. Leakage of facial information will pose a serious threat to users' privacy. Therefore, the data is authenticated before being uploaded to the cloud and encrypted to ensure security during transmission.
- (2) The concealability of attributes in access policy. In the traditional CP-ABE scheme, the access policy is embedded in the ciphertext. It is required that the attributes in the set of attributes owned by the visitor can satisfy the attributes in the access structure, so as to decrypt the data. In fact, regardless of whether the decryption is successful or not, some important information can be deduced according to the

existing plaintext access policy. Therefore, in order to eliminate the security risks caused by plaintext transmission of the access policy, the attribute values in the access policy can be encrypted and hidden.

- (3) Confirmability of target data deletion in the cloud. Users usually store their data in the cloud. However, the cloud is honest but curious. It may be driven by interest to extract some useful information and leak it to the analysis organization. Users do not want their information to be permanently stored in the cloud. When they want to delete data in the cloud, the cloud may be reluctant to delete or fraudulently delete it for hidden business interests, but users cannot verify whether their data has actually been deleted. Therefore, it is particularly important to delete the target data with assurance and confirmability.

In the process of facial recognition and resolution, the framework offloads some tasks from the cloud to the edge server by applying the task partitioning strategy, rather than performing all the facial resolution processes in the cloud. It makes full use of the powerful computing and parsing capabilities of the edge server, which not only significantly reduces the amount of personal information transmission, but also reduces the computing pressure of cloud. However, how to ensure the data security of transmission, access, and stored procedures needs to be solved urgently. Thus, we propose a cloud data verifiable deletion scheme in response to the above security issues to ensure the security of the scheme.

4. Verifiable Deletion Scheme of Cloud Data in Facial Recognition and Resolution

In this section, we first optimize the MTI session key agreement scheme, which ensures the correctness of channel transmission by confirming the identity between the sender server and the receiver server. In addition, based on the analysis of the security and privacy issues of the framework, we introduce the verifiable deletion scheme of cloud data in detail.

4.1. Optimized MTI Session Key Agreement Scheme. To ensure the security of the channel during transmission, the identity between the resolution server and the management server needs to be verified. Firstly, we optimize the MTI session key agreement scheme, which has the ability to resist replay attacks and parallel sessions. The symbol is shown in Table 1.

The system first exposes q and g to RS and MS. RS has a unique $ID_{RS}, x_{RS}, y_{RS} = g^{x_{RS}} \bmod q$, authorization $C_{RS} = (ID_{RS}, y_{RS}, Sig_{TA}(ID_{RS}, y_{RS}))$ certificate

$$C_{RS} = (ID_{RS}, y_{RS}, Sig_{TA}(ID_{RS}, y_{RS})), \quad (1)$$

which binds public key and identity.

Similarly, MS has a unique $ID_{MS}, x_{MS}, y_{MS} = g^{x_{MS}} \bmod q$, authorization certificate

$$C_{MS} = (ID_{MS}, \gamma_{MS}, Sig_{TA}(ID_{MS}, \gamma_{MS})), \quad (2)$$

which binds public key and identity.

RS randomly selects $r_{RS} \in [0, q-1]$, then calculates

$$S_{RS} = g^{r_{RS}} \text{mod} q, \quad (3)$$

and sends $\{S_{RS}, ID_{MS}, C_{RS}\}K_{MS}$ to MS.

MS randomly selects $r_{MS} \in [0, q-1]$ and then calculates

$$s_{MS} = g^{r_{MS}} \text{mod} q. \quad (4)$$

$$K_{RS,MS} = s_{RS}^{x_{MS}} \gamma_{RS}^{r_{MS}} \text{mod} q. \quad (5)$$

The system gets $K_{RS,MS}$ which is the session key for further communication between RS and MS. Then, it sends $\{S_{MS}, ID_{RS}, C_{MS}\}K_{RS}$ to RS.

RS gets

$$K'_{RS,MS} = s_{MS}^{x_{RS}} \gamma_{MS}^{r_{RS}} \text{mod} q, \quad (6)$$

by calculation and sends $\{S_{RS,MS}, ID_{MS}, C_{RS}\}K_{MS}$ to MS. MS judges whether $K'_{RS,MS}$ is consistent with $K_{RS,MS}$ or not. If $K'_{RS,MS} = K_{RS,MS}$, it indicates that the negotiation is successful, and RS and MS have completed authentication between each other. If $K'_{RS,MS} \neq K_{RS,MS}$, the negotiation fails.

In this algorithm, the shared key can be derived from the (q, s_{MS}) or (g, s_{RS}) , but not from (s_{MS}, s_{RS}) . In other words, although the attacker can eavesdrop on q, g, s_{RS}, s_{MS} and even ciphertext, it cannot export the correct session key $K_{RS,MS}$ due to the unknown values of r_{RS} and r_{MS} . Therefore, it cannot crack the ciphertext.

The algorithm adds a fresh factor every time the message is sent and binds the source and destination of the message, which can effectively prevent replay attacks. Because the session keys (S_{RS}, S_{MS}) are randomly selected, attackers can only destroy the formation of the key but has no way to launch a parallel session attack against it. The optimized MTI session key agreement scheme defines the authorization certificate and increases the authentication process, which improves the security of the scheme.

4.2. Encryption Scheme in Facial Recognition. Firstly, there are two communication channels that are absolutely safe. One is the channel among the trusted authority (TA), RS, and MS, and the other is the channel between the client and RS. Secondly, MS uses read-only access to decrypt files and will not tamper with relevant data. Furthermore, the communication channel between RS and MS is not secure, and the cloud storage center is also semi-honest.

When the user collects the facial image, the edge server generates facial data through facial detection, facial image preprocessing, feature extraction, and facial identifier generation algorithms. MTI session key agreement scheme and SHA-1 hash algorithm are used to ensure the security and integrity of data transmission, and CP-ABE algorithm is used to encrypt personal data and fine control access as shown in Figure 2. The symbol definition and description list of facial recognition and resolution are in Table 2.

The encryption scheme in facial recognition is as follows.

4.3. Verifiable Deletion Scheme of Cloud Data in Face Resolution. In the process of facial resolution, the client first collects the facial image of the person being tested, and the edge server resolves it into a facial identifier and uploads it to the cloud. The cloud server first uses the decryption algorithm to decrypt the ciphertext file and then matches the decrypted facial identifier with the facial identifier uploaded in the data center. The serial number of personal information bound with the facial identifier is returned to the edge server to match with each other. The verifiable deletion scheme in face resolution is as follows:

When the user completes the facial resolution, some information in the cloud is no longer available. If user needs to delete it, the verifiable deletion scheme can be used to cancel the user's access to facial data. Our scheme is to add a verifiable process after the user deletes the data to ensure the deletion succeeds completely, and also to avoid false deletion of the cloud. This enables users to better control their own data, and the security of data in the cloud is effectively ensured as shown in Figure 3.

5. 5. Experimental Simulation Results Analysis

5.1. Security Analysis. This part mainly analyzes the security of data transmission in the process of facial recognition and resolution, the concealability of access policy attributes, and the confirmability of cloud data deletion.

- (1) In our security scheme, we adopt the MTI session key agreement scheme. In the process of session key agreement, the public keys of RS and MS are allocated to be shown in public. They perform bidirectional authentication to confirm each other's identity and ensure antireplay attack by adding fresh factors and randomly selecting keys each time they send messages. At the same time, we use the SHA-1 algorithm. All the facial identifiers $(K_{RS,MS}, CT || sig_{ssk}R) || Hash(CT || sig_{ssk}R)$ stored in the data center are extracted and decrypted; we calculate $Hash'(CT || sig_{ssk}R)$ and $Hash(CT || sig_{ssk}R)$ to ensure the integrity of the data access process and effectively prevent malicious tampering by illegal users.
- (2) The scheme first generates the symmetric key DK through AES symmetric key algorithm and construct the access policy by the CP-ABE algorithm. Then, it uses DK to encrypt the data, which is further encrypted to the ciphertext CT associated with the access policy A . Based on this, the plaintext attribute values in the access policy are successfully partially hidden. Visitors must make their own attributes meet the access policy attributes in order to achieve the access and decryption of the data.
- (3) In the cloud storage environment, when the data owner wants to delete the outsourced data, in order to avoid logical deletion, the attribute access control policy

TABLE 2: Symbol definition list of facial recognition and resolution process.

Term	Description	Page
I	Original facial image obtained from the client	8
INF	Registered personal information	8
V	The facial identifier generated by the original facial image	8
SN	Serial number parsed from personal information	8
I'	The original facial image of the subject	8
V'	Facial identifier of the subject	8

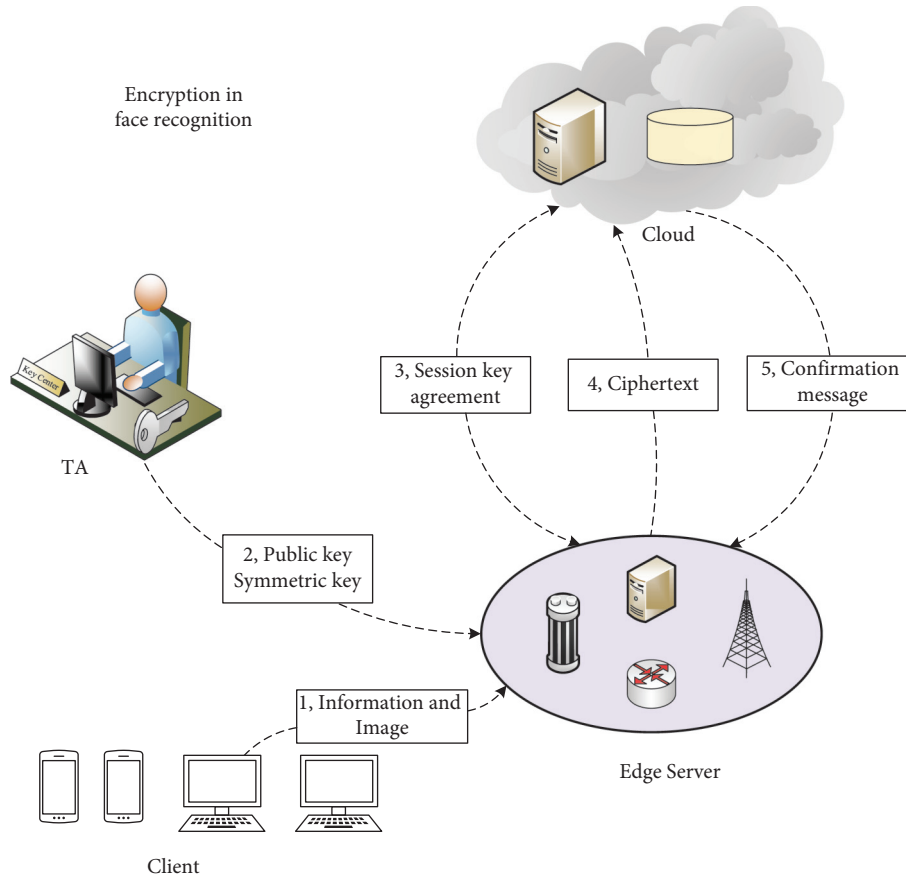


FIGURE 2: Face recognition process.

corresponding to the ciphertext is changed by re-encrypting the ciphertext to achieve fine-grained operation and deterministic permanent deletion. In the proposed scheme, we use CP-ABE algorithm to generate the reencryption key and then use it to encrypt the ciphertext CT to generate new ciphertext and MHT tree roots. RS compares the new and old tree root. And if the two are equal, it indicates that the target data has indeed been completely deleted in the cloud.

5.2. Setup of Experiment. In this part, we add the facial recognition and resolution security privacy protection scheme based on edge computing to the prototype system and then verify the security and effectiveness of the scheme through a large number of simulation experiments. Because

the facial recognition process is similar to the facial resolution process, we only test the effectiveness of verifiable deletion schemes in facial resolution.

This experiment uses one cloud server, two edge servers, and six mobile terminal devices to build the system as shown in Figure 1. All the algorithms are tested on a Win10-64-bit laptop using Intel i7-6700HQ processor at 2.60 GHz. We use the following three face databases: Caltech face image database, GT face image database, and BioID face image database. These three databases are, respectively, composed of 450 color face images of 27 characters, 750 face avatars of 50 characters, and 1521 gray-scale face images of 23 characters. We firstly preprocess the original database and randomly select 50 sheets from them. Then, the preprocessed data is used as the database for our experiment.

Begin

Step 1: TA inputs the security parameter γ to generate the public key PK and the master key MK of the CP-ABE algorithm. TA inputs the security parameter λ to generate the symmetric key DK by the AES symmetric key algorithm. Then, TA assigns PK and DK to RS.

Step 2: RS inputs attribute set U to construct access policy A by CP-ABE attribute encryption algorithm. RS uses the symmetric key DK of the AES algorithm to encrypt the data (V, SN) , which binds the face identifier and the serial number. Then, TA generates data ciphertext $DE K$. Based on the public key PK and the access policy A , RS uses the CP-ABE algorithm to encrypt the data ciphertext $DE K$ into the ciphertext CT associated with A . Then, ciphertext data CT generates the signature $sig_{ssk}R$. R is the root of the constructed MHT, and ssk is the signature private key.

Step 3: The session key agreement algorithm between RS and MS is executed according to Algorithm 1. It verifies the identity of FN and MS and generates the session key $K_{RS,MS}$.

Step 4: RS firstly uses the SHA-1 hash algorithm to calculate $Hash(CT\|sig_{ssk}R)$, and FN uploads $(K_{RS,MS}, CT\|sig_{ssk}R)\|Hash(CT\|sig_{ssk}R)$ to MS.

Step 5: After MS receives the message, it calculates the $Hash'(CT\|sig_{ssk}R)$ for comparison with $Hash(CT\|sig_{ssk}R)$. If $Hash'(CT\|sig_{ssk}R) = Hash(CT\|sig_{ssk}R)$, it has the data integrity and sends $(K_{RS,MS}, CT\|sig_{ssk}R)\|Hash(CT\|sig_{ssk}R)$ to data center. Finally, it returns the successful registration flag.

End

ALGORITHM 1: The encryption scheme in facial recognition.

Begin

Step 1: TA inputs the master key MK in Algorithm 1 and the attribute set ψ corresponding to data user DU . It outputs the private key SK of the CP-ABE algorithm and sends the private key SK to MS.

Step 2: According to Algorithm 1, RS and MS perform authentication between each other. The session key agreement scheme is also performed to generate the session key $K_{RS,MS}$.

Step 3: RS uses to SHA-1 hash algorithm to calculate $Hash(V'\|SK)$, and then RS uploads $(K_{RS,MS}, V'\|SK)\|Hash(V'\|SK)$ to MS. MS calculates $Hash(V'\|SK) = Hash'(V'\|SK)$ to check the data integrity and get face identifier V' and SK .

Step 4: MS receives the CT stored in the data center. It inputs the private key SK and ciphertext CT and uses the CP-ABE algorithm to decrypt. The data ciphertext $DE K$ is obtained. Then, it inputs DK and the ciphertext $DE K$ through the AES algorithm to obtain the decrypted data (V, SN) .

Step 5: MS uses the face identifier matching algorithm to match V' and V successfully to obtain (V, SN) . It uses hash algorithm to calculate $Hash(V'\|SN)$ and then returns $(K_{RS,MS}, V'\|SN)\|Hash(V'\|SN)$ to RS.

Step 6: RS calculates $Hash'(V'\|SN)$ to judge the data integrity. Then, it obtained personal information INF by matching the serial number SN and returned it to the client to complete the face resolution.

Step 7: TA inputs RS's deleted request RR and the master key MK and generates reencryption key PK' through CP-ABE algorithm. Then, TA returns it to RS and then to MS.

Step 8: MS inputs the ciphertext CT and the reencryption key PK' . MS outputs the reencrypted ciphertext CT' and the new MHT tree root \tilde{R} through the CP-ABE algorithm. Then, it returns \tilde{R} to RS.

Step 9: RS receives \tilde{R} and compares it with R . If $R \oplus \tilde{R} = 0$, that is to say, R is equal to \tilde{R} , it proves that the cloud server has indeed successfully deleted the target data.

End

ALGORITHM 2: Verifiable deletion scheme of cloud data in face resolution.

5.3. Analysis of Experimental Results and Performance

5.3.1. Network Transmission Volume. In order to evaluate the impact of the communication overhead of the scheme in practical application, we mainly measure the network transmission from the edge server to the cloud server. Compared with the traditional scheme, this scheme can reduce the amount of data transmission. In this experiment, three face databases are tested. Figure 4 shows an increase of only 0.132 kb relative to the prototype framework system and a decrease of 0.044 kb relative to the original security framework system. The experimental results show that the experimental communication overhead is relatively small, which can meet the needs of practical applications and have good stability.

5.3.2. Response Time of Facial Database. We include the time of issuing the request, generating the facial identifier, network transmission, data encryption and decryption, identifier matching, and verifiable deletion of the target data in the cloud. Compared with security scheme, our scheme increases the computing time of data encryption and decryption and data verifiable deletion. Through comparing it with the experiment without face database, Figure 5 shows that the average consumption of this scheme increases by 84 ms milliseconds compared to the system without security framework and only increases by 14 ms compared to the system with security framework. But we can safely manage data in the cloud, which shows that our experiments can satisfy the practical application to a certain extent.

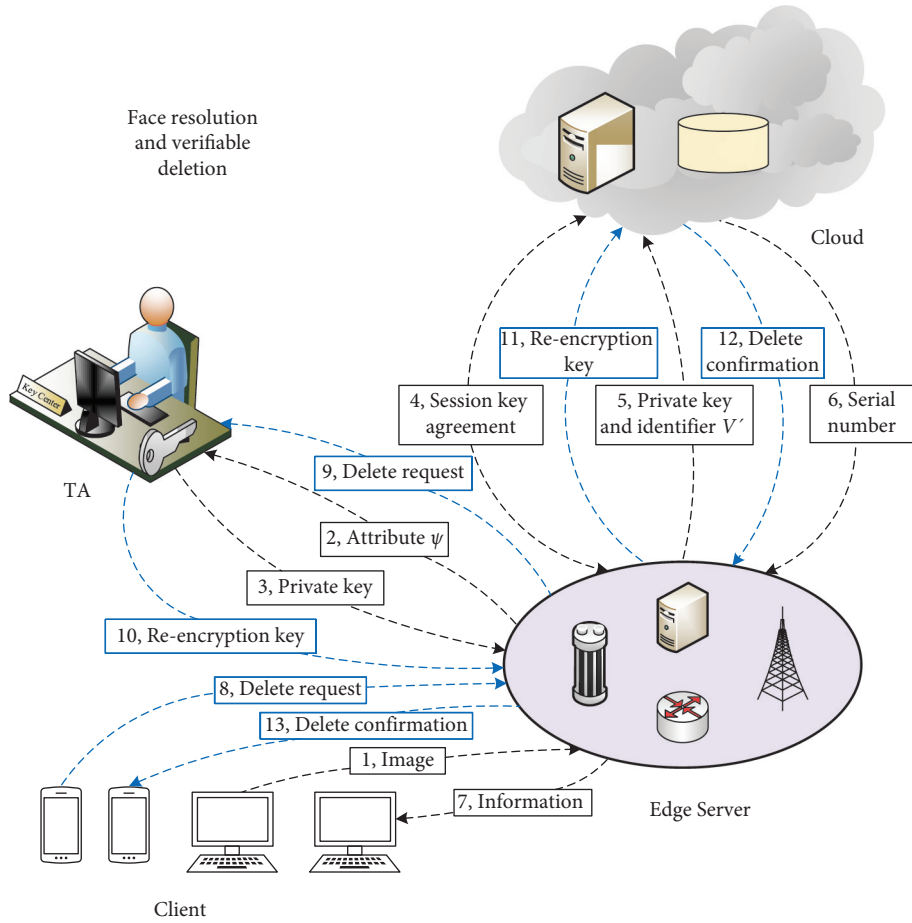


FIGURE 3: Verifiable deletion process of cloud data in face resolution.

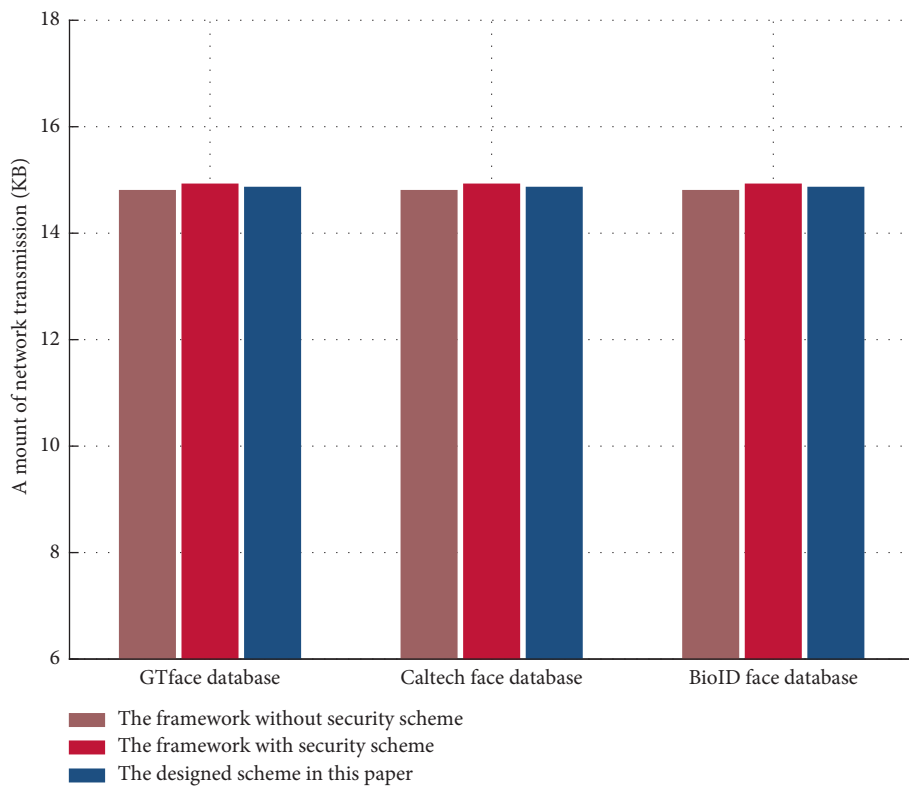


FIGURE 4: Amount of network transmission for different face databases.

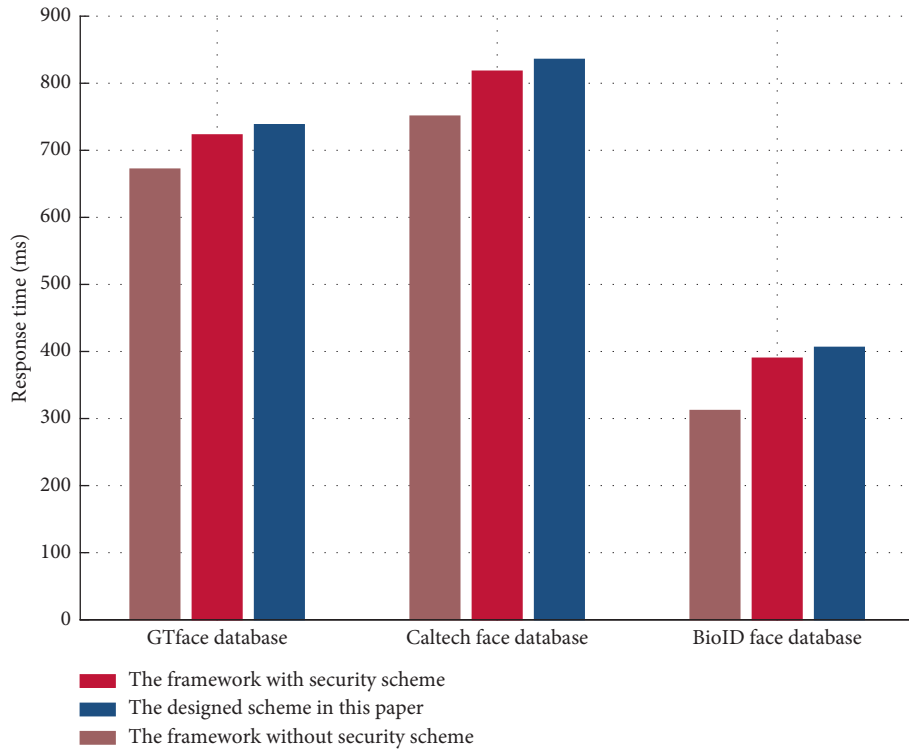


FIGURE 5: Average response time for different face databases.

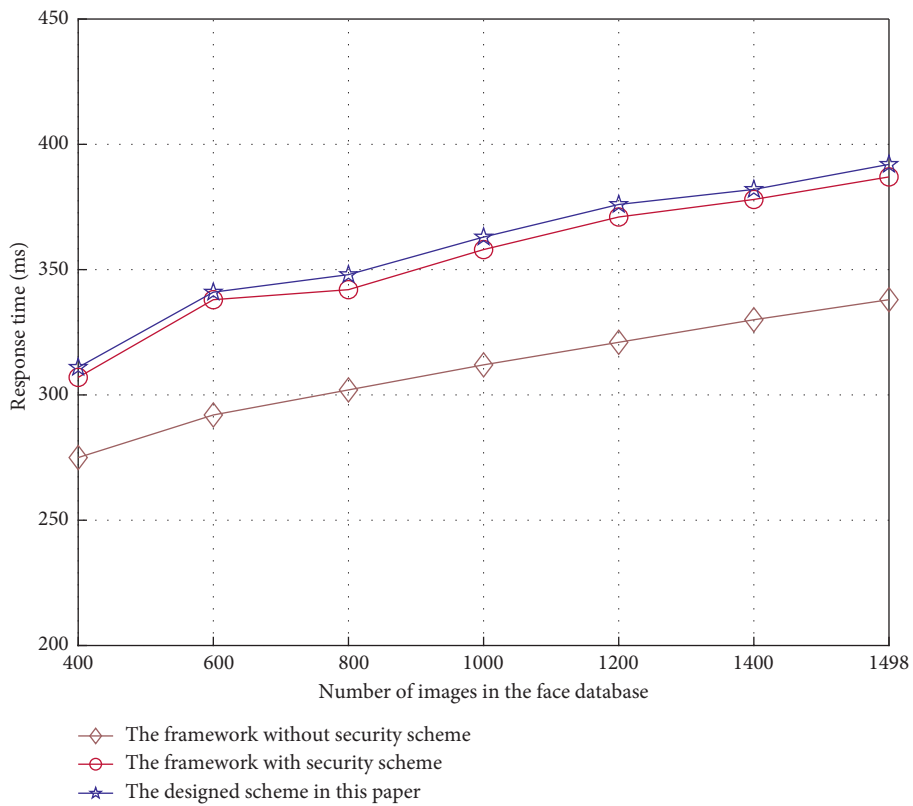


FIGURE 6: Average response time for different size of face databases.

5.3.3. *Response Time of Face Databases of Different Sizes.* In order to better reflect the performance of the experiment, we use BioID face database to test the response time of the system. By selecting the face database test in the range of 400–1600, Figure 6 shows that the scheme grows steadily with the increase of face database size, without great instability. The time consumption of this scheme increases by 64 ms compared to the system without security framework and increases by 12 ms compared to the system with security framework. This shows that our scheme has good advantages in stability.

6. Conclusion

In this paper, we focused on the privacy security of facial recognition and resolution framework based on edge computing. In summary, we analyzed the security threats of facial recognition and resolution framework, including the security of data transmission, the concealability of access policy attributes, and the verifiability of cloud data deletion. To solve these problems, we improved the framework by combining the characteristics of cloud computing and edge computing. To further ensure the security of cloud data transmission and storage, we proposed a verifiable deletion scheme based on Hidden CP-ABE, which can effectively prevent attackers from stealing sensitive information and deleting data falsely. Then, we applied this scheme to the facial recognition and resolution framework based on edge computing and evaluated its performance by simulation experiments. The results indicated that the proposed scheme performs good stability and can effectively meet the requirements of facial recognition and resolution in practical application. In future work, we will further verify this scheme through experiments in more dimensions. Moreover, on the premise of ensuring the efficiency of facial recognition and resolution, the performance of low energy consumption and low latency performance will be optimized at a higher level.

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.

References

- [1] S. Ramesh and M. Govindarasu, “An efficient framework for privacy-preserving computations on encrypted IoT data,” *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8700–8708, 2020.
- [2] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, “Psspr: a source location privacy protection scheme based on sector phantom routing in wsns,” *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1204–1221, 2022.
- [3] T. Li, Y. Chen, Y. Wang et al., “Rational protocols and attacks in blockchain system,” *Security and Communication Networks*, vol. 2020, pp. 1–11, 2020.
- [4] A. Naouri, H. Wu, N. A. Nouri, S. Dhelim, and H. Ning, “A novel framework for mobile-edge computing by optimizing task offloading,” *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 13065–13076, 2021.
- [5] G. Li, J. Cao, J. Wu, X. Ren, and H. Yu, “Dimension reduction algorithm based on adaptive maximum linear neighborhood selection in edge computing,” *IEEE Internet of Things Journal*, vol. 11, no. 19, pp. 2327–4662, 2020.
- [6] X. Xu, R. Mo, X. Yin et al., “P. D. M.: Privacy-aware deployment of machine-learning applications for industrial cyber-physical cloud systems,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5819–5828, 2020.
- [7] Y. Shin, D. Koo, J. Yun, and J. Hur, “Decentralized server-aided encryption for secure deduplication in cloud storage,” *IEEE Transactions on Services Computing*, vol. 13, no. 6, pp. 1021–1033, 2020.
- [8] K. Gai, K. Xu, Z. Lu, M. Qiu, and L. Zhu, “Fusion of cognitive wireless networks and edge computing,” *IEEE Wireless Communications*, vol. 26, no. 3, pp. 69–75, 2019.
- [9] P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, and K. Chen, “Privacy-preserving outsourced classification in cloud computing,” *Cluster Computing*, vol. 21, no. 1, pp. 277–286, 2018.
- [10] G. Li, Y. Liu, J. Wu, D. Lin, and S. Zhao, “Methods of resource scheduling based on optimized fuzzy clustering in fog computing,” *Sensors*, vol. 19, no. 9, pp. 21–22, 2019.
- [11] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [12] H. Ning, X. Liu, X. Ye, J. He, W. Zhang, and M. Daneshmand, “Edge computing-based ID and nID combined identification and resolution scheme in IoT,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6811–6821, 2019.
- [13] K. Yu, J. Yu, X. Cheng, D. Yu, and A. Dong, “Efficient link scheduling solutions for the Internet of Things under Rayleigh fading,” *IEEE/ACM Transactions on Networking*, vol. 29, no. 6, pp. 2508–2521, 2021.
- [14] M. Barbieri, K. T. P. Nguyen, R. Diversi, K. Medjaher, and A. Tilli, “RUL prediction for automatic machines: a mixed edge-cloud solution based on model-of-signals and particle filtering techniques,” *Journal of Intelligent Manufacturing*, vol. 32, no. 5, pp. 1421–1440, 2021.
- [15] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, “Semi-selfish mining based on hidden Markov decision process,” *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
- [16] X. Zhao, J. Wu, M. Wang, G. Li, H. Yu, and W. Feng, “Multi-sensor data fusion algorithm based on adaptive trust estimation and neural network,” in *Proceedings of the 2020 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 582–587, Chongqing, China, August 2020.
- [17] K. Yu, B. Yan, J. Yu, H. Chen, and A. Dong, “Methods of improving secrecy transmission capacity in wireless random networks,” *Ad Hoc Networks*, vol. 117, Article ID 102492, 2021.
- [18] Z. Ma, Y. Liu, X. Liu, J. Ma, and K. Ren, “Lightweight privacy-preserving ensemble classification for face recognition,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5778–5790, 2019.
- [19] J.-H. Im, S.-Y. Jeon, and M.-K. Lee, “Practical privacy-preserving face authentication for smartphones secure against malicious clients,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2386–2401, 2020.

- [20] Y. Chen, S. Dong, T. Li, Y. Wang, and H. Zhou, "Dynamic multi-key fhe in asymmetric key setting from lwe," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5239–5249, 2021.
- [21] S. Qi, Y. Lu, W. Wei, and X. Chen, "Efficient data access control with fine-grained data protection in cloud-assisted IIoT," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2886–2899, 2020.
- [22] Y. Zhang, J. Li, and H. Yan, "Constant size ciphertext distributed CP-ABE scheme with privacy protection and fully hiding access structure," *IEEE Access*, vol. 7, pp. 47982–47990, 2019.
- [23] H. Tian, X. Li, H. Quan, C.-C. Chang, and T. Baker, "A lightweight attribute-based access control scheme for intelligent transportation system with full privacy protection," *IEEE Sensors Journal*, vol. 21, no. 14, pp. 15793–15806, 2020.
- [24] Y. Yu, L. Guo, S. Liu, J. Zheng, and H. Wang, "Privacy protection scheme based on CP-ABE in crowdsourcing-IoT for smart ocean," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10061–10071, 2020.
- [25] C. Ge, Z. Liu, J. Xia, and L. Fang, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1214–1226, 2021.
- [26] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, 2021.
- [27] L.-Y. Yeh, P.-Y. Chiang, Y.-L. Tsai, and J.-L. Huang, "Cloud-based fine-grained health information access control framework for lightweightiot devices with dynamic auditing andattribute revocation," *IEEE transactions on cloud computing*, vol. 6, no. 2, pp. 532–544, 2015.
- [28] M. Miao, J. Wang, J. Ma, and W. Susilo, "Willy. Publicly verifiable databases with efficient insertion/deletion operations," *Journal of Computer and System Sciences*, vol. 86, pp. 49–58, 2017.
- [29] H. Ma, R. Zhang, S. Sun, Z. Song, and G. Tan, "Server-aided fine-grained access control mechanism with robust revocation in cloud computing," *IEEE Transactions on Services Computing*, p. 1, 2019.
- [30] K. Edemacu, B. Jang, and J. W. Kim, "Collaborative ehealth privacy and security: an access control with attribute revocation based on OBDD access structure," *IEEE journal of biomedical and health informatics*, vol. 24, no. 10, pp. 2960–2972, 2020.
- [31] G. Yu, X. Zha, X. Wang et al., "Enabling attribute revocation for fine-grained access control in blockchain-IoT systems," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1213–1230, 2020.

Research Article

ATMChain: Blockchain-Based Security Framework for Cyber-Physics System in Air Traffic Management

Xin Lu  and Zhijun Wu 

School of Safety Science and Engineering, Civil Aviation University of China, Tianjin, China

Correspondence should be addressed to Zhijun Wu; zjwu@cauc.edu.cn

Received 16 November 2021; Revised 24 December 2021; Accepted 14 February 2022; Published 10 March 2022

Academic Editor: Yuling Chen

Copyright © 2022 Xin Lu and Zhijun Wu. 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 air traffic management (ATM) system is an intelligent system that integrates the ground computer network, the airborne network, and the space satellite (communication and navigation) network. It has remarkable characteristics of cyber-physical system (CPS). The development of ATM system is inseparable from the application of new technologies. This paper proposes a security framework based on blockchain for CPS in the ATM system. Through the research on the characteristics of blockchain and CPS, this paper analyzes the necessity of integrating them into the research of the ATM system, demonstrates the feasibility of combining them, and then constructs the ATMChain framework mechanism to realize the in-depth integration of blockchain and CPS in ATM. On this basis, this paper gives the overall design architecture and implementation steps of the scheme. In addition, this paper also makes a series of analysis and demonstration from the perspective of scheme security. The research scheme will help to improve the security, reliability, and scalability of ATM services and provide a new reference for establishing a more security and efficient ATM system.

1. Introduction

As a large-scale key infrastructure for the safe operation of civil aviation, the air traffic management (ATM) system includes a large number of heterogeneous functional subsystems, which constitutes a typical complex system [1]. With the continuous improvement of the informatization and networking degree of the ATM system, the interaction between its components is closer, the data in the system need to be highly shared, and the functions of each subsystem need to support each other, which has the typical characteristics of the cyber-physical system (CPS) [2]. The CPS is a comprehensive system that involves computer algorithms and cyber and physical objects. The CPS monitors and perceives the objects of the physical world and controls the behavior of physical entities by mining and analyzing the rich data contained in the physical world, so as to realize the efficient operation of the physical world [3, 4]. As a typical CPS complex system, the interwound systems of systems have brought great security pressure to ATM systems all

over the world because of its complexity [5]. In order to deal with and solve the security risks in the ATM system, experts and scholars in academic and aviation circles began to introduce mature and widely applied new technologies, such as microservice, cloud computing, edge computing, big data, Internet of things, artificial intelligence, and blockchain, into ATM field, providing reference and ideas for the further development of civil aviation informatization [6, 7]. It will promote the construction and development of the ATM system.

At this stage, the concept of CPS has become one of the core guiding ideology adopted by Federal Aviation Administration (FAA) in deploying NextGen, the third generation ATM system [8, 9]. It boldly changes the traditional mode of ATM system deployment and more deeply implements the concept of safe and green system construction integrating human, machine, environment, and management. Secondly, for the typical CPS system, some relevant scholars have carried out research from the perspective of blockchain distributed architecture to eliminate some

potential security risks in the CPS system [10–14]. Finally, relevant researchers consider using blockchain technology to solve the related security problems faced in the ATM system [15–17]. However, there is no research on the integration of CPS and blockchain in the ATM system. Therefore, combined with relevant research, taking the ATM system as the research background and from the perspective of CPS, this paper integrates the distributed architecture idea and principle of blockchain into the security framework research of the ATM system, so as to solve the bottleneck problems existing in the current theoretical research, promote technological innovation, and promote the application of basic research. Furthermore, it can also contribute to promoting the development and construction goal of “smart ATM” of International Civil Aviation Organization (ICAO).

As a highly informative industry, civil aviation has always attached great importance to information security and information value transmission [18]. Coincidentally, blockchain technology has greater value in both of these areas. The characteristics of the blockchain provide possibilities for its application in the field of ATM. This paper is committed to building a CPS security architecture based on blockchain for the ATM system. The structure of the paper is as follows. The second part describes the research status of the ATM system, blockchain, and CPS. The third part gives the CPS security architecture in ATM based on blockchain and describes the scheme in detail from three aspects: the scheme background, scheme framework, and the information sharing algorithm. The fourth part analyzes the security of the scheme framework proposed in the previous part. The fifth part provides the conclusion and prospect and discusses the next research direction and focus.

2. Background

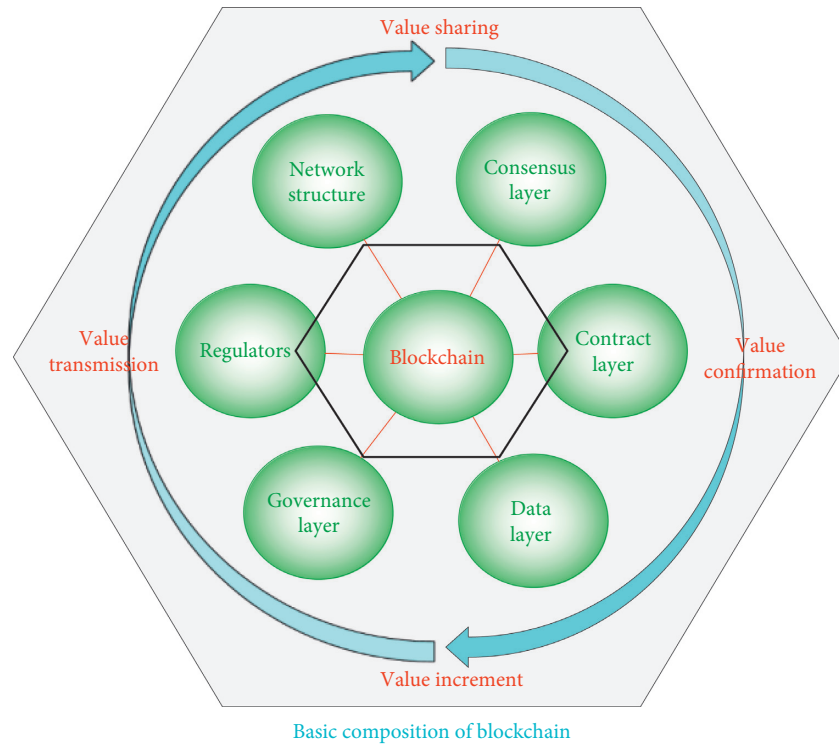
This part starts from two aspects. On the one hand, it introduces the essence of blockchain and CPS. On the other hand, as for the special research background of the ATM system, it points out the feasibility of using blockchain technology from the perspective of CPS in ATM.

2.1. Blockchain and Cyber-Physical System. Blockchain technology, which appeared in 2008, is a decentralized, tamper proof, forgeable, and collectively maintained distributed database management method [19]. It is a new computer technology application mode integrating distributed data storage, point-to-point transmission, consensus mechanism, and encryption algorithm (Figure 1(a)). In this paper, the blockchain technology is regarded as the integration and addition of multiple technologies, a security concept of information system, rather than a specific technology. From the perspective of data recording, blockchain is a chain data structure formed by connecting and combining data blocks in sequence according to time, which ensures its tamper ability and unforgeability as a distributed ledger in a cryptographic way [20]. Bookkeeping is accompanied by the development history of human society. The evolution process of bookkeeping form is from

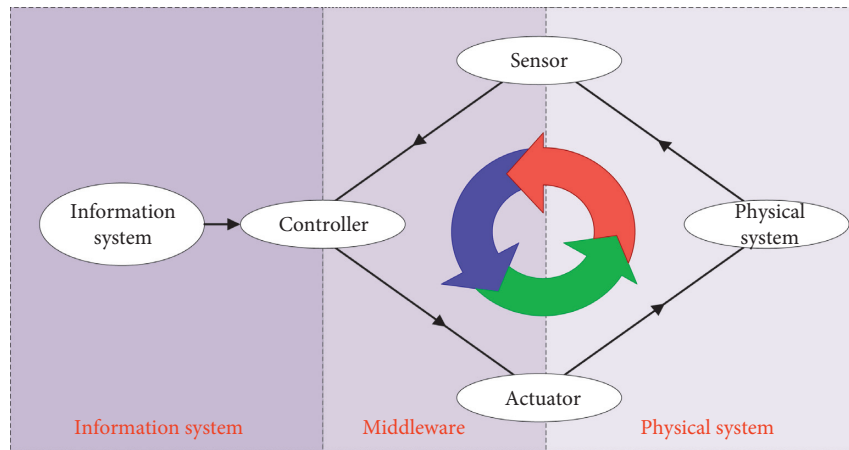
single ledger to double ledger and then to digital ledger. At present, it has developed to distributed ledger. Blockchain technology is a representative technology of distributed ledger. The core of blockchain technology is the deposit-certificate and value-certificate, as well as the consensus mechanism formed on this basis, that is, the algorithm to reach a consensus on the sequence of things over a period of time. Furthermore, in the whole blockchain system, various application services based on information flow are realized in the form of smart contract, which endows “information” with value and reconstructs the current centralized social organization production relationship. When the data that can reflect the facts are fixed, value is generated from it. Data are the means of production in the era of digital economy, and the algorithm is a productivity tool for data processing. The blockchain adds a dimension of trust to the existing data, improves the flow efficiency, and completes the reconstruction of decentralized production relations.

By combing the relevant theoretical viewpoints, it is not difficult to find that CPS uses perception, communication, network, and other technical means to realize the perception and digital presentation of the physical world, so as to form the data projection in the information world (as shown in Figure 1(b)). Then, the advanced computer algorithm is used to optimize the operation process of the physical world, and a closed loop of mutual influence and interaction between the actual entity world and the cyber world is formed. Its key core features are integration, diversity, and intelligence. The CPS is a complex system with close integration of the physical system and information system. It can be divided into information system, physical system, and middleware, which advances the close interaction between actual system and information system. Different from the traditional independent physical system and information system, the CPS emphasizes the strong interaction between them. The development trend of close integration has led to a leap of quality in the development of physical systems. By means of the computing and storage capacity of information systems for large-scale information, the level of intelligence, automation, and systematization of physical systems has been further improved. The middleware is the key component to promote the integration of CPS, including controller, sensor, and actuator. The controller is a component for data pre-processing and instruction forwarding. It plays a control function and closely couples the information system with the middleware. Sensor is an important input unit for collecting physical system state data, which provides important data support for CPS [21–23]. The actuator is the driving unit that acts on the physical system. The sensor and actuator closely couple the physical system with the middleware.

2.2. Air Traffic Management. The ATM system is a large CPS system integrating space-based network, air-based network, and land-based network (as shown in Figure 2). Its information system is composed of ground control station and data fusion center, which can generate intelligent decision results through the comprehensive analysis of aviation data and meteorological data. The intermediate components



(a)



(b)

FIGURE 1: (a) Basic composition of blockchain. (b) Basic composition of CPS.

include a sensor network composed of ADS-B (Automatic Dependent Surveillance-Broadcast) sensors, radars, satellites, and so on and an actuator network composed of aircraft and so on. Its physical systems include aircraft and airports. The various parts of the whole ATM system are deeply integrated with the continuous improvement of its automation level. Accordingly, the ATM system also contains some subsystems that can be regarded as CPS. For example, the aircraft is a small CPS system. Its information system is composed of decision-making units of the airborne avionics system. The intermediate part includes airborne sensor unit and actuator unit, including sensors such as

aircraft state information acquisition and environmental parameter acquisition and actuators that drive changes in flight course and speed. Its physical system includes aircraft engine and fuselage. The subsystems of the whole aircraft are highly coordinated, which enhances the intelligence of flight with the support of information. In short, the ATM system is a complex system with the characteristics of wide spatial distribution, complex functions, time delay sensitivity, and high security requirements. It is a large CPS as a whole, including some small CPSs. Therefore, the ATM system is the CPS of CPSs, which has the characteristics of extremely obvious information physical system.

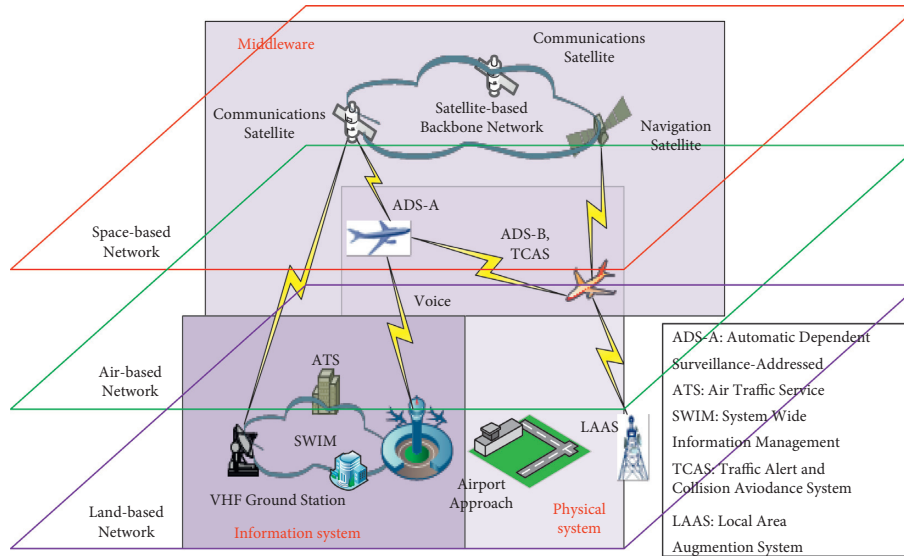


FIGURE 2: Basic composition of the ATM system.

The ATM plays an important role in the sustainable development of aviation safety because it is directly related to the efficiency and safety of air transportation [24–28]. However, when building ATM networks, software engineering designers generally do not pay enough attention to information security. For instance, in the case of the public unencrypted broadcast protocol ADS-B, any external party can eavesdrop, tamper, and delete ADS-B information with relative ease [16, 29]. At the same time, the widely distributed composition and the multisource heterogeneous data of the ATM system have a natural unity with the decentralized and distributed bookkeeping characteristics of blockchain. For the ATM, how to design a reliable and secure information delivery and sharing approach by using blockchain technology and combining the CPS features of ATM is the main problem addressed in this paper. Next, the feasibility is discussed from three aspects:

- (1) In the development of the ATM system, the biggest problem faced by the cooperation between ATM departments all over the world, even in different regions of a single country, is security and privacy protection. At present, the main threats faced by the ATM system are data leakage, data deception, entity camouflage, and denial of service (DoS). These security threats are distributed in the information system, middleware, and physical system of the ATM system. At present, many research studies are to solve the challenges from the traditional security strategies and methods. The emergence of blockchain technology provides a useful tool to solve the trust problem of human society, and it has become one of the primary application technologies to solve the issues of security and privacy protection.
- (2) According to the different object-oriented of blockchain, it includes three kinds: public chain, private chain, and federated chain. Public blockchain is open to all, and nodes can join at will. Private blockchain is only open to individual entities, such as the interior of a

company or organization. Alliance blockchain will be open to a specific industry organization. Alliance blockchain refers to a blockchain in which several institutions participate in bookkeeping; that is, industry alliance members reach a consensus through trust in multicenters. This feature is just suitable for the distributed and multicenter network of the ATM system. In addition, compared with the public blockchain, the very important feature of the alliance blockchain is the node access control and national security standard support. This ensures that authenticated access and regulatory rules are developed in compliance with regulatory requirements and increases the speed of transactions based on trusted security. The security framework of this paper is designed based on alliance blockchain.

- (3) From the current research results of system design, the architecture of the distributed system can better meet the requirements of users for system robustness and controllability than the central architecture. Therefore, the obvious research trend is to establish a strong and secure ATM framework from the viewpoint of multilevel overall system layout design, combined with the concept of CPS. As a representative decentralized information storage solution, blockchain coincides with such a research trend, and it is also necessary to apply it to ATM.

3. The Design of Security Framework

Starting from the business characteristics of the ATM system, combined with the concept of blockchain and CPS, this paper proposes an ATM-CPS security architecture based on alliance blockchain, abbreviated as ATMChain. The following describes the ATMChain security framework from three aspects: scheme background, scheme framework, and the information sharing algorithm.

3.1. The Background of ATMChain. With the rise of intelligent transportation technology all over the world, civil aviation is developing in the direction of information sharing, structure, and function dependence through 3C (computing, communication, and control) technology. Among them, the ATM system has the characteristics of typical CPS. Using CPS modelling theory to model the ATM system can fully analyse the interactions between its cyber system and physical system. It can also make up for the one sidedness of the existing research after separating the information system and the actual system and enhance the pertinence of system analysis. At the same time, the decentralized framework adopted by the blockchain provides a novel view for ATM infrastructure layout optimization. The basic level of the ATM system takes the sensing device as the physical carrier to store its function information and environment sensing information. The ground control station is responsible for processing the information delivered by ATM equipment at each basic level. Here, these information processing centers will serve as the nodes of the alliance blockchain, promote the interconnection of ATM information in the whole region, and provide value data services to ATM users on this basis. This distributed architecture can effectively prevent the dysfunction of local ATM nodes, so that the global ATM system can run efficiently and securely. In addition, blockchain technology's chain construction method and decentralized storage provide a novel way to implement information governance traceability. Therefore, in order to better achieve the goal of "smart ATM," two challenges still need to be met.

- (1) CPS digitally presents and intelligently manages the real ATM system. A lot of reliable information in the physical world will be stored in the information system. Once the system is attacked, it will not only cause information leakage but also cause a large number of actual parties to be utilized by attackers, resulting in great damage to enterprises and society. The traditional information system storage model adopts the centralized storage mode. Once the central system is attacked, the whole system will face the disaster of destruction.
- (2) The integration of multiple technologies has become a challenge for the development of the ATM system. Internet of things (IoT) is the data source, big data is the basic resource, cloud computing is the infrastructure, and artificial intelligence is the core algorithm. Blockchain creates conditions for the transformation of ATM business infrastructure and operation mechanism. The CPS is a comprehensive technical system based on automatic data flow between information space and physical space, including state perception, real-time analysis, scientific decision-making, and accurate execution. The comprehensive application, cross support, and virtuous iteration of cyclic evolution of these new technologies will actively promote the development of ATM intelligence.

3.2. The Framework of ATMChain. In the ATM system, there is often a lack of corresponding encryption technology in the key information delivery link, which makes data tampering and privacy disclosure a major threat. This article presents a security framework to protect the rights and interests of all participants in the information flow of the ATM system. The scheme is designed based on the HyperLedger Fabric of the alliance blockchain architecture. In this paper, ATM is separated into basic physical layer, information processing layer, and information delivery layer. ATMChain framework based on the principle of blockchain distributed architecture is proposed to optimize the layout of ATM-CPS and enhance the overall robustness of the system. Integrating blockchain technology with ATM at all layers and establishing a robust and dependable cyber system will significantly enhance the information security of ATM and optimize the layout of the information system.

3.2.1. System Model. The system model diagram of the ATM-CPS security framework implemented by alliance blockchain technology is shown in Figure 3. Three parties are comprised in the system model: ATM user, ATM information processing entity, and ATM physical space entity. This paper adopts the PBFT [30] consensus mechanism which is most commonly used in alliance blockchain network. Since the consensus mechanism is not the focus of this paper, it will not be described in detail. Their functions are described as follows:

ATM User. It requests ATM message service, purchases information resources, and pays relevant fees.

ATM Information Processing Entity. As a full node operation in the ATMChain, it performs access control and information authorization through smart contracts [31]. It has the characteristics of intelligence and is the main carrier for executing ATM services.

ATM Physical Space Entity. It operates as a light node in ATMChain and encapsulates the information and data storage into the whole node. In addition, the node also collects various requests from the upper information processing entity.

3.2.2. System Framework. This paper focuses on the "value object" of "ATM information" and studies the ATMChain system framework, from information collection to information right confirmation and release, to information transmission, sharing, and use, and finally to the value-added feedback of information value. The intelligent ATM-CPS framework is constructed by tracing the records of the whole life cycle of ATM information. The ATMChain contains three types of nodes. One is the light node responsible for information collection in the ATM physical layer, and the other is the whole node that stores and processes all information and blocks. The third node is the ATM user, and this type of node does not participate in the task of block consensus and block storage. Therefore, ATMChain architecture can be divided into three levels [32–36].

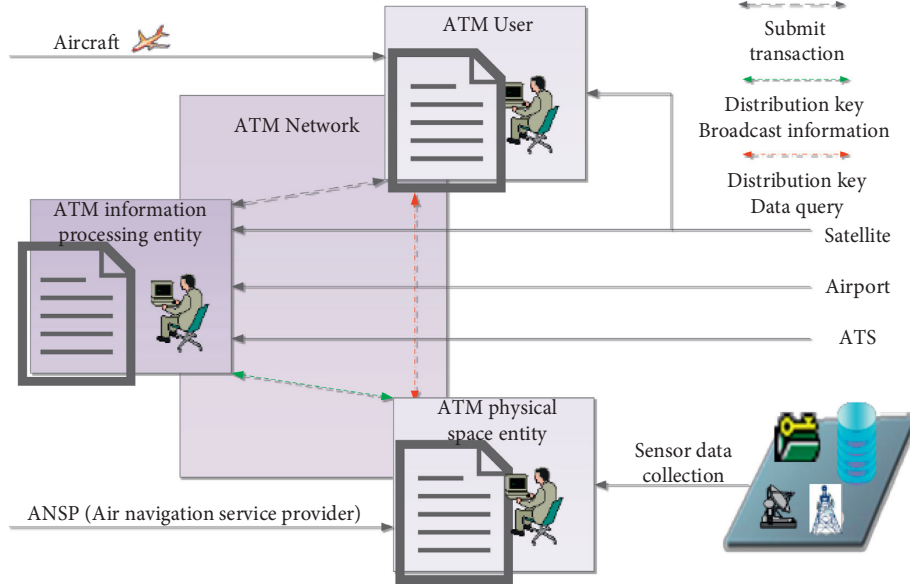


FIGURE 3: System model diagram of ATMChain.

Firstly, as the bottom support of the whole ATMChain, the CPS sensor physical layer equipment at the basic level not only has the basic sensing, information storage, and transfer functions of traditional CPS physical components but also needs to have the functions of information standardization, mutual data transmission information, and backup of transmission records. Secondly, the whole node of the integration level composed is based on the IoT that perceives the physical world. The nodes in this layer are not only required to achieve deep aggregation and interoperability of the ATMChain base layers, as well as the transfer of heterogeneous information, the allocation of response, speed and frequency of information query requests, but also to audit and authenticate the newly accessed CPS clients in the base layers. In addition, it is its task to unify information format or establish heterogeneous information transformation protocols, unify local scope linkage protocols, maintain block information of this layer, and record device linkage traces and so on. Finally, ATM users constitute the application layer of ATMChain architecture.

The ATMChain system scheme mainly includes six steps: system initialization stage, ATM information collection and uplink stage, ATM information release and authorization stage, ATM information sharing and use stage, block generation stage, and system consensus stage. The operation view of ATMChain system is shown in Figure 4. The definitions of some symbols are given as shown in Table 1. The steps of the system are described as follows.

Step 1. System initialization stage: At this stage, blockchain nodes will establish a blockchain-based ATM information sharing framework through the ECDSA signature algorithm and public key cryptography system. ATM users register in the ATMChain through KYC (know your customer) mechanism and their real identity. Among them, the user's key pair, certificate, and wallet address are $(pk_i, sk_i, cert_i, \text{and } WID_i)$, respectively. $Cert_i$ can only use

the bound registration information to identify the user. According to the ECDSA algorithm, this scheme uses secure elliptic curve parameters, including curve $E_p(a, b)$ and base point G , then the U_i selects private key sk_i , and uses G to calculate public key pk_i as shown in the following formula:

$$\begin{cases} sk_i = k (k < n) \\ pk_i = kG, \end{cases} \quad (1)$$

U_i sends its wallet address WID_i to a third partner, which generates $(PKI, sk_i, cert_i, WID_i)$. When U_i runs system initialization, the wallet address used is selected from the nearest node account pool. After selection, U_i needs to check the integrity of the wallet and obtain the details. Among them, the account pool stores all transaction records. In addition, the key pair of ATM information processing node is $(sk_B, pk_B) = (k', k'G)$.

Step 2. ATM information collection and uplink stage: At this stage, the light nodes in the ATM physical space collect ATM data into the ATM information processing entity through various sensors. Before transmitting ATM information upward, these light nodes should standardize their own physically perceived data and realize consistent authentication by means of digital signature.

Step 3. ATM information release and authorization stage: The processed ATM information is officially released to the ATMChain with the signature of publisher M . At this time, the specific content of ATM information is encrypted. If you want to use ATM information, you need to obtain M 's authorization. All nodes back up and broadcast the published ATM information to each other, so that a wider range of ATM users can use it.

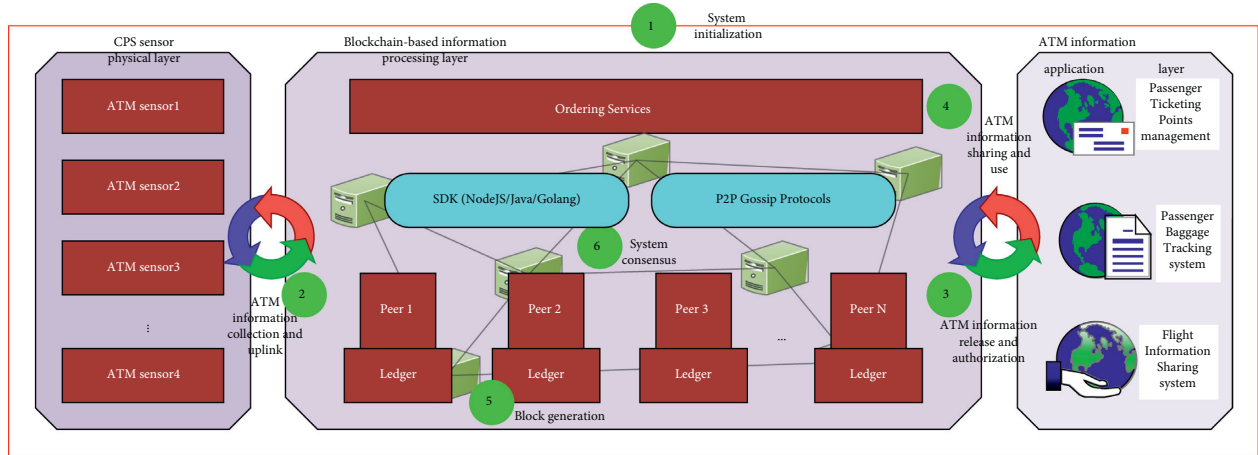


FIGURE 4: The operation view of the ATMChain system.

TABLE 1: Symbol definition of ATMChain.

Symbol	Definition
U_i	The ATM user of ATMChain
M	The ATM information processing entity
$Pk_{i/B}, sk_{i/B}$	The public and private keys of U_i or M
$cert_i$	Certificate of identity
WID_i	Wallet address of user with ID
$Hash$	The hash value of m
δ_i	User's signature in authorization information
t_1, t_2	The time stamp
$-1, 0, 1$	Status of information transaction
M	The transaction information

Step 4. ATM information sharing and use stage: At this stage, ATM users interact with ATM information processing node M by running a series of signature algorithms to complete the authorization of ATM information use. The specific process is that U_i sends a request for information authorization to nodes, and then nodes broadcast the message to M . M needs to provide timely feedback within a specified period of time. After receiving the feedback, nodes match U_i 's request authorization information with M to complete the transaction.

Step 5. Block generation stage: Before running ATMChain, the nodes participating in the block consensus have been selected to join the system. In order to accomplish the target of more "distributed," the more consensus nodes, the stronger the robustness of the system. In particular, to ensure the authenticity and accuracy of the information, the node will collect all local transaction records and encrypt or sign them at each specific time, packaging them to generate ATM information transaction blocks. These transaction records construct new blocks that refer to the hash value of the previous block, and they will be stored in ATMChain in chronological order. Such a process ensures the traceability of the whole life cycle of ATM information transaction. At this point, the block generation phase ends.

Step 6. System consensus stage: At this stage, the nodes use the PBFT consensus algorithm to reach consensus, thus maintaining the reliable performance of the system. At present, the consensus mechanisms commonly used in alliance blockchain include Pool verification-pool and Byzantine fault-tolerant algorithm. The former is based on traditional distributed consistency technology and information verification mechanism. It can realize second level consensus verification and is suitable for multiparty Multi Center Alliance blockchain. The latter belongs to the state machine Byzantine protocol, which reduces the complexity of the algorithm from exponential level to polynomial level. The PBFT is a consensus algorithm implemented and recommended by HyperLedger Fabric. It adopts the scheme of "one node one vote" to determine the accounting results, with good performance. It is mainly used in alliance blockchain.

3.3. The Information Sharing Algorithm. The ATM information sharing is a crucial step in the whole ATMChain. The specific process is shown as follows:

Step 1. The U_i runs $ECDSA.UserSign$ algorithm and enters his key pair (sk_i, pk_i) and relevant parameters, and then the algorithm will output the signature δ_i . The specific steps are as follows:

- (a) Generate a random integer d ($d < n$ and n is the order of G), and calculate R and r according to the following formula:

$$\begin{cases} R(x, y) = dG, \\ r = x \bmod n. \end{cases} \quad (2)$$

- (b) The coordinate values of point $R(x, y)$ and ATM information m are set as parameters, and the hash value and s are calculated by formula (3) using the hash function SHA256:

$$\begin{cases} \text{Hash} = \text{SHA256}(m, x, y), \\ s = (\text{Hash} + rk)d^{-1} \bmod n. \end{cases} \quad (3)$$

The signature $\delta_i = (s, d)$, and x and $\text{Hash}(m)$ should be rounded up.

Step 2. U_i sends $\{\text{cert}_i, pk_i, \delta_i, m\}$ to M and generates the information use authorization request by the following formula:

$$\text{req}_i = \{\text{cert}_i, pk_i, \delta_i\}. \quad (4)$$

Step 3. M needs to verify the received information after receiving the request. If cert_i exists or authentication fails, the request is rejected. Of course, if cert_i does not exist and the verification is successful, the request is accepted. The specific verification steps are shown in the following formulas:

- (a) First, calculate the following:

$$\text{Hash} = \text{SHA256}(m, x, y), \quad (5)$$

$$\begin{cases} u = s^{-1} \text{Hash}(m) \bmod n, \\ v = s^{-1} r \bmod n, \\ (x', y') = uG + vpk_i = uG + v(kG), \\ r' = x' \bmod n. \end{cases} \quad (6)$$

x and $\text{Hash}(m)$ should be rounded up.

- (b) Verify according to the following formula:

$$\begin{aligned} &? \\ &r = r'. \end{aligned} \quad (7)$$

If equation (7) is satisfied, the message can be accepted; otherwise, it is invalid. After successful verification, M will accept the request and store the data $(\text{cert}_i, pk_B, pk_i, 1, 0)$ in the local account pool; among them, 1 represents the status of valid transaction and 0 represents the status of newly generated transaction that have not been transferred. In addition, -1 represents the status of transaction awaiting transfer.

Step 4. M signs $(\text{enroll}, \text{cert}, pk_B, pk_i, t_1)$ by running $\text{ECDSA.MerchantSign}$ algorithm (similar to ECDSA.UserSign algorithm) and using the private key sk_B , where t_1 represents the ATM information request time and enroll represents the information of U_i . Then, the $(\delta_i, \text{request}_i)$ is sent to the blockchain. The specific signing steps are as follows.

According to formulas (8) and (9), M runs the algorithm and signs m_i :

$$\begin{cases} R(x, y) = d_1G, \\ r_1 = \bar{x} \bmod n, \end{cases} \quad (8)$$

$$\begin{cases} \text{Hash}_1 = \text{SHA256}(m_1, x, y), \\ s_1 \equiv d_1^{-1} (\text{Hash}_1 + kr_1) \bmod n, \end{cases} \quad (9)$$

where $m_1 = \{\text{enroll}, \text{cert}_i, pk_B, pk_i, t_1\}$ and x and $\text{Hash}(m_1)$ should be rounded up. The final signature is calculated as $\delta_i = (s_1, r_1)$. Then, U_i verifies the signature according to equations (5)–(7).

4. Security Analysis and Simulation

The ATM system is a security sensitive system, and its security objectives are consistent with those of other computer information systems, such as information confidentiality, integrity, availability, and traceability. Based on the research results in the academic field, this paper describes the security of ATM-CPS as security threat, system vulnerability, security attack, and security measures. Meanwhile, the security concerns are decomposed into security mechanisms and security objectives. Security measures refer to the measures to build a secure and robust ATM by integrating security mechanisms and security objectives. The ATMChain framework based on blockchain technology can be regarded as ATM security measures. Therefore, this paper will analyse the scheme from four dimensions: ATM information confidentiality, ATM information integrity analysis, ATM information availability, and ATM information traceability. Finally, the scheme is simply simulated from the perspective of communication cost.

4.1. Information Confidentiality Analysis of ATM. The first thing to measure the security of an information system is to ensure the confidentiality of information, and ATMChain is no exception. Information confidentiality refers to hiding information or resources. Confidentiality means that even if unauthorized persons or organizations are aware of the existence of information resources, they cannot obtain them. ATMChain integrates the principle of blockchain with CPS to give full play to the security mechanism of blockchain. In the blockchain, the data between nodes are backed up synchronously to ensure that the ATM participants entering the system share information. At the same time, the information is encrypted by the encryption algorithm and transmitted through asymmetric key pairs, which greatly improves the information confidentiality in the process of ATM information transmission.

4.2. Information Integrity Analysis of ATM. Information integrity refers to the credibility of data or resources, which is usually used to prevent improper modification of data or unauthorized tampering of data. Loss of integrity means that the information is subject to unauthorized tampering and information loss. In ATMChain, M can prove the

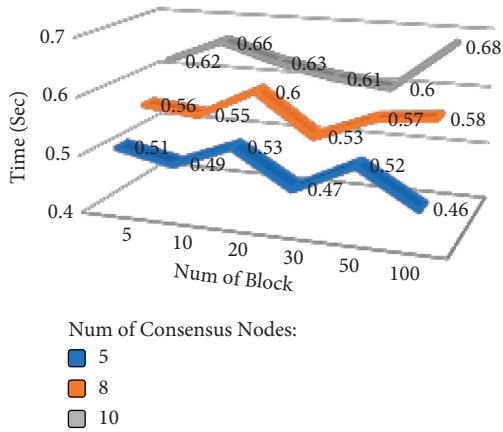


FIGURE 5: The generation time of block.

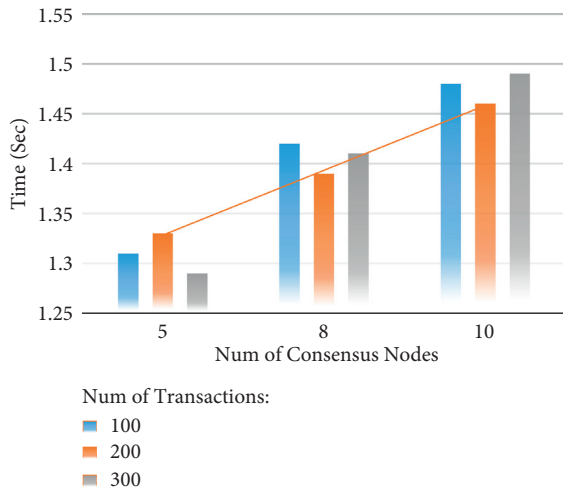


FIGURE 6: The time of transaction confirmation.

correctness of the signature of the authorized user and the authorization to request ATM information through the following formula. Therefore, when M verifies successfully, it indicates that the signature and the authorization information are legal and correct. Then, M will accept the request and store the information $(cert_i, pk_B, pk_i, 1, 0)$ in the local account pool and continue to execute the next algorithm. The verification mechanism avoids the deceptive attack of illegal nodes disguised as authorized nodes in central authorization. In addition, after the data blocks are added to the blockchain, each consensus node will also verify the block data. Only the verified data blocks can be added to the blockchain to guarantee the authenticity of the data. Last but not least, the unique tamper proof characteristics of the blockchain also greatly increase the stability of the data in the block.

4.3. Information Availability Analysis of ATM. Information availability refers to the ability to use the required information or resources. Loss of availability means that access to or use of information or information systems is

blocked. Denial of service (DoS) is one of the most threatening security attacks faced by the ATM system. Therefore, the availability goal in ATMChain is to prevent DoS attacks to ensure the availability of information. Meanwhile, a remarkable feature of the ATM system is the integration of time and space, so ensuring the real-time operation is also an important aspect. Under the ATMChain framework, the blockchain distributed architecture can realize the interactive loop between the actual world and the cyber world, which is different from the traditional single centralized control, avoids the DoS attack of the attacker on the system, and ensures the availability of information. In ATMChain, the hysteresis caused by information perception, transmission, control, and optimization feedback is also compensated by the full sharing advantage of information to ensure real-time and field tracking.

4.4. Information Traceability Analysis of ATM. Information traceability is an important information security goal in addition to confidentiality, integrity, and availability. Information traceability means that information can be tracked. In ATM, traceability means real-time tracking of the operation of physical entities. When there is a deviation in the operation state of the physical world, the time and cause of the deviation can be accurately found. In the ATMChain, each block includes two parts: block header and block body. The block header encapsulates the current block header value, preblock header hash value, timestamp, random number, and other ATM information. By encapsulating the hash value of the front block in each block, the current block is connected with its front block to form a chain structure. The sequence of blocks in the blockchain is confirmed by the time sequence stamped with time stamps, and it is consistent with the historical sequence of time stamps. Thus, the blockchain structure with time sequence is formed. The data are arranged in chronological order to ensure the historical traceability of the information. When there is a deviation in the operation state of the actual entity, the error information fed back to the ATM system can be found according to the chronological order. Meanwhile, the information in the basic layer CPS can be stored in the ATM information processing layer, so the deviation can be traced from the information processing layer.

4.5. Simulation. In terms of the communication cost of this scheme, that is, the interaction time of the whole algorithm during ATM information sharing, the main influencing factors are the transaction confirmation time and the block generation time. The scheme runs experiments on a genuine Intel computer device, using the programming language Python, the operating system is windows 10, CPU 2.5 GHz, and running memory 16 GB. We are concerned with the time required for ATM information sharing transactions to be confirmed and recorded on the blockchain. There are two signatures in the transaction confirmation process, one for U_i and one for M . There are also two signatures for verification.

In this paper, we first test the block generation time in ATMchain to test its throughput. Each experiment randomly generates a number of blocks (averaged as block generation time), and each block contains 5 transactions. As shown in Figure 5, under the condition that the number of consensus nodes is constant, the block generation time is independent of the frequency of initiated transactions, while with the increase in the number of consensus nodes, the block generation time also increases slightly, but all can meet the actual deployment requirements. Second, the time for transaction confirmation might be affected by the frequency of transactions and the number of consensus nodes under the condition that there are no errors in two message verifications. The results show that only the number of consensus nodes has an impact on the transaction confirmation time (as in Figure 6). Therefore, the relationship between the number of consensus nodes and the throughput needs to be balanced to obtain the best system performance. Finally, with a consensus node of 5, it can be estimated that the process of a single information sharing transaction from generating to recording on the blockchain takes about 1.82 s in total, which can meet the actual demand.

5. Conclusion

This paper presents a blockchain and CPS integration architecture for the ATM system. The purpose is to grasp the CPS characteristics of the ATM system, integrate the advantages of blockchain technology, promote the research and development, and improve the service processing capacity of existing ATM. In terms of ideas, firstly, after summarizing and analyzing the characteristics and research background, it leads to the benefits of the research thinking of blockchain and CPS to the development of the ATM system. Then, taking ATMChain security architecture as the core, this paper details the research background, framework, and key algorithm. Based on the traceability and nontamperability of blockchain in data storage and sharing, the secure sharing of multiparty heterogeneous data in ATM environment is designed. Finally, the security of ATMChain framework is analysed from four dimensions, which are confidentiality, integrity, availability, and traceability of information security. This paper provides a novel useful idea for the construction, development, and research of the ATM system.

In the following research, firstly, the degree of decentralization of architecture design can still be further optimized, and the distributed subsystem can be used for cluster management. Secondly, it will be proposed to further optimize and refine the research design of ATMChain through the research on the consensus mechanism and smart contract of the blockchain. In addition, in terms of system security and privacy, we can integrate zero knowledge proof, homomorphic encryption, and other technologies with the system to realize data encryption and identity concealment, improve the privacy of the system, as well as the design of multilevel identity authentication and access control [37–39]. Finally, the concept of interaction and integration of human, machine, environment, and management, which

has been emphasized in ATM system research, is still lacking in ATMChain design, which is also the direction of future efforts.

Data Availability

This paper is an article on the design of air traffic management architecture. At present, it has only carried out preliminary theoretical analysis and research and has not carried out experimental simulation based on actual data. In future research, the corresponding data sets can be shared.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the joint funds of the National Natural Science Foundation of China and Civil Aviation Administration of China (U1933108), the Scientific Research Project of Tianjin Municipal Education Commission (2019KJ117), and the Tianjin Research Innovation Project for Postgraduate Students (2021YJSB240).

References

- [1] L. Bogoda, J. Mo, and C. Bil, “A systems engineering approach to appraise cybersecurity risks of CNS/ATM and avionics systems,” *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, ICNS, in *Proceedings of the 2019 Integrated Communications, Navigation and Surveillance Conference*, pp. 1–15, April 2019.
- [2] K. Sampigethaya and R. Poovendran, “Aviation cyber-physical systems: foundations for future aircraft and air transport,” *Proceedings of the IEEE*, vol. 101, no. 8, pp. 1834–1855, 2013.
- [3] K. Sampigethaya and R. Poovendran, “Cyber-physical integration in future aviation information systems,” in *Proceedings of the 2012 IEEE/AIAA Thirty First Digital Avionics Systems Conference (DASC)*, pp. 7C2-1–12, Williamsburg, VA, USA, October 2012.
- [4] W. Zhang, M. Kamgarpour, D. Sun, and C. J. Tomlin, “A hierarchical flight planning framework for air traffic management,” *Proceedings of the IEEE*, vol. 100, no. 1, pp. 179–194, 2012.
- [5] ICAO, “Cybersecurity Strategy,” 2019, <https://www.icao.int/cybersecurity/Pages/Cybersecurity-Strategy.aspx>.
- [6] M. Shengdong, X. Zhengxian, and T. Yixiang, “Intelligent traffic control system based on cloud computing and big data mining,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6583–6592, 2019.
- [7] R. Sabatini, A. Roy, E. Blasch et al., “Avionics systems panel research and innovation perspectives,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 12, pp. 58–72, 2020.
- [8] M. Mitici and H. A. P. Blom, “Mathematical models for air traffic conflict and collision probability estimation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1052–1068, March 2019.
- [9] P. Park and C. Tomlin, “Investigating Communication Infrastructure of Next Generation Air Traffic Management,” in *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, pp. 35–44, Beijing, China, April 2012.

- [10] A. Kanak, N. Ugur, and S. Ergun, "Diamond Accountability Model for Blockchain-Enabled Cyber-Physical Systems," in *Proceedings of the 2020 IEEE International Conference on Human-Machine Systems (ICHMS)*, pp. 1–5, Rome, Italy, September 2020.
- [11] A. Gu, Z. Yin, C. Fan, and F. Xu, "Safety framework based on blockchain for intelligent manufacturing cyber physical system," in *Proceedings of the 2019 First International Conference on Industrial Artificial Intelligence (IAI)*, pp. 1–5, Shenyang, China, July 2019.
- [12] W. Zhao, C. Jiang, H. Gao, S. Yang, and X. Luo, "Blockchain-enabled cyber-physical systems: a review," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4023–4034, 2021.
- [13] 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.
- [14] M. Y. Afanasev, Y. V. Fedosov, A. A. Krylova, and S. A. Shorokhov, "An Application of Blockchain and Smart Contracts for Machine-To-Machine Communications in Cyber-Physical Production Systems," in *Proceedings of the 2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pp. 13–19, St. Petersburg, Russia, May 2018.
- [15] M. Dehez Clementi, N. Larriue, E. Lochin, M. A. Kaafar, and H. Asghar, "When Air Traffic Management Meets Blockchain Technology: A Blockchain-Based Concept for Securing the Sharing of Flight Data," in *Proceedings of the 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, pp. 1–10, San Diego, CA, USA, September 2019.
- [16] F. Hasin, T. H. Munia, N. N. Zumu, and K. A. Taher, "ADS-B based air traffic management system using ethereum blockchain technology," in *Proceedings of the 2021 International Conference on Information and Communication Technology for Sustainable Development*, pp. 346–350, ICICT4SD), Dhaka, Bangladesh, February 2021.
- [17] I. S. Bonomo, I. R. Barbosa, L. Monteiro et al., "Development of SWIM registry for air traffic management with the blockchain support," in *Proceedings of the 2018 Twenty First International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3544–3549, Maui, HI, USA, November 2018.
- [18] A. Sternstein, "Exclusive: FAA computer systems hit by cyberattack earlier this year," 2015, <https://www.nextgov.com/cybersecurity/2015/04/faa-computer-systems-hit-cyberattack-earlier-year/109384/>.
- [19] Y. Yuan and F. Wang, "Blockchain: the state of the art and future trends," *Acta Automatica Sinica*, vol. 42, no. 4, pp. 481–494, 2016.
- [20] China Blockchain Technology and Industry Development Forum, "China blockchain technology and application development white paper," China Blockchain Technology and Industry Development Forum, China, (in Chinese), 2016.
- [21] Z. Wu, T. Shang, and A. Guo, "Security issues in automatic dependent surveillance - broadcast (ads-B): a survey," *IEEE Access*, vol. 8, Article ID 122147, 2020.
- [22] X. Koutsoukos, G. Karsai, A. Laszka et al., "SURE: a modeling and simulation integration platform for evaluation of secure and resilient cyber-physical systems," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 93–112, 2018.
- [23] B. Besselink, V. Turri, S. H. van de Hoef et al., "Cyber-physical control of road freight transport," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1128–1141, 2016.
- [24] ICAO, *Global ATM Operational Concept*, International Civil Aviation Organization, Montreal, Canada, Doc I. 9854, 2005.
- [25] ICAO, *Manual on ATM Requirements*, International Civil Aviation Organization, Montreal, Canada, Doc I. 9882, 2008.
- [26] ICAO, *Manual on Collaborative Air Traffic Flow Management*, International Civil Aviation Organization, Montreal, Canada, Doc I. 9965, 2012.
- [27] ICAO, *Global Air Navigation Plan*, International Civil Aviation Organization, Montreal, Canada, Doc I. 9750, 4 edition, 2013.
- [28] Y. Wu, X. Lu, and Z. Wu, "Blockchain-based trust model for air traffic management network," in *Proceedings of the 2021 IEEE Sixth International Conference on Computer and Communication Systems*, pp. 92–98, ICCCS), Chengdu, China, April 2021.
- [29] Y. Chen, S. Dong, T. Li, Y. Wang, and H. Zhou, "Dynamic multi-key FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5239–5249, 2021.
- [30] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer PBFT consensus for blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1146–1160, 2021.
- [31] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F. Wang, "An overview of smart contract: architecture, applications, and future trends," in *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 108–113, Changshu, China, June 2018.
- [32] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, pp. 3596–3612, 2021.
- [33] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "A source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, vol. 37, pp. 1–18, 2021.
- [34] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, vol. 36, pp. 1–22, 2021.
- [35] X. Lu, Z. Wu, Y. Wu, Q. Wang, and Y. Yin, "ATMChain: Blockchain-Based Solution to Security Problems in Air Traffic Management," in *Proceedings of the 2021 IEEE/AIAA Fortyth Digital Avionics Systems Conference (DASC)*, pp. 1–8, San Antonio, TX, USA, October 2021.
- [36] X. Lu and Z. Wu, "ATMCC: design of the integration architecture of cloud computing and blockchain for air traffic management," in *Proceedings of the 2021 IEEE International Symposium on Parallel and Distributed Processing with Applications*, pp. 37–43, New York City, NY, USA, October 2021.
- [37] B. S. Egala, A. K. Pradhan, V. Badarla, and S. P. Mohanty, "Fortified-chain: a blockchain-based framework for security and privacy-assured Internet of medical things with effective access control," *IEEE Internet of Things Journal*, vol. 8, no. 14, Article ID 11717, 2021.
- [38] S. Rathore and J. H. Park, "A blockchain-based deep learning approach for cyber security in next generation industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5522–5532, 2021.
- [39] Z. Zhou, B. Wang, M. Dong, and K. Ota, "Secure and efficient vehicle-to-grid energy trading in cyber physical systems: integration of blockchain and edge computing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 43–57, 2020.

Research Article

Privacy-Preserving Minority Oversampling Protocols with Fully Homomorphic Encryption

Maohua Sun , Ruidi Yang , and Mengying Liu 

School of Management and Engineering, Capital University of Economics and Business, Beijing 100070, China

Correspondence should be addressed to Maohua Sun; sunmaohua@cueb.edu.cn

Received 18 November 2021; Accepted 28 January 2022; Published 10 March 2022

Academic Editor: Yuling Chen

Copyright © 2022 Maohua Sun 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, blockchain and machine-learning techniques have received increasing attention both in theoretical and practical aspects. However, the applications of these techniques have many challenges, one of which is the privacy-preserving issue. In this paper, we focus on, specifically, the privacy-preserving issue of imbalanced datasets, a commonly found problem in real-world applications. Built based on the fully homomorphic encryption technique, this paper presents two new secure protocols, Privacy-Preserving Synthetic Minority Oversampling Protocol (PPSMOS) and Borderline Privacy-Preserving Synthetic Minority Oversampling Protocol (Borderline-PPSMOS). Our analysis reveals that PPSMOS is generally more efficient in performance than Borderline-PPSMOS. However, Borderline-PPSMOS achieves a better TP rate and F-Value than PPSMOS.

1. Introduction

In the past few years, new information technology techniques, such as blockchain [1–4] and machine-learning [5–15], have been developing rapidly and used successfully in various real-life applications. However, they still face a critical challenge in the privacy-preserving issue. For example, the openness of a blockchain system poses a serious threat to the privacy and security of any user transactions. Thus, research for privacy-preserving techniques is becoming even more crucial.

Datasets in the wild come with a variety of problems. One of the most common problems is the imbalanced issue of the datasets. Imbalanced datasets issue arises in many real-world sectors, such as disease detection [16], bankruptcy prediction [17], fraud detection [18], etc. As the distribution of samples is incorrect in imbalanced datasets, it may cause the classification algorithms to produce inaccurate results and further issues. An imbalanced dataset usually consists of a number of classes, which falls into one of these two types: majority classes, which has a bigger number of examples, and minority classes, in which there are fewer examples. In this paper, we consider the situation where there are only two classes in a dataset, i.e., one majority class and one minority class.

The existing solutions proposed to solve the imbalanced dataset problem are categorized according to which level the technique is solving the problem from, e.g., data level, feature level, and machine-learning algorithm level. In this paper, we focus on fixing the problems at the data level. There are two known data level techniques, namely undersampling and oversampling methods. The undersampling method works by removing parts of the samples from the majority class to balance the ratio of majority and minority samples, whereas oversampling method balances the majority and minority samples by generating new minority samples. In 1972, Wilson [19] proposed an undersampling method, in which a majority sample should be deleted if all of its neighbors are minority samples. In 2020, Wang et al. [20] proposed a novel entropy and confidence-based undersampling boosting framework to solve imbalanced dataset issues, which could be applied to noniterating algorithms such as decision trees.

Random oversampling of minority classes is the simplest oversampling method. Through sampling with replacement, samples are continuously drawn from the minority class. This method, however, can easily lead to data overfitting. In 2002, Chawla et al. [21] proposed the Synthetic Minority Oversampling Technique (SMOTE) algorithm, which is one

of the best-known oversampling methods to date. The algorithm works by generating artificial data using bootstrapping and the K-nearest neighbor algorithm. Further improvising SMOTE algorithm, Han et al. [22] proposed Borderline-SMOTE in 2005. The algorithm focuses on working on samples that are on the boundary of both majority and minority classes. The demonstration showed that Borderline-SMOTE achieved a better TP rate and F-Value than its predecessor. In 2008, Douzas et al. [23] presented a simple and effective oversampling method based on K-means clustering and SMOTE, which is able to eliminate noise generation and effectively overcome imbalances between and within classes. Furthermore, Li et al. [24] presented three sampling approaches for imbalanced learning in 2020. Unlike the previous solutions, their approaches considered a new class-imbalance metric, which contains the differences of information contents between classes, instead of the traditional imbalance ratio.

Although so many solutions have been proposed to solve the imbalanced data sets problem, the privacy-preserving issue has not been well resolved. To the best of our knowledge, Hong et al. [25] proposed a secure collaborative machine-learning solution in which they used secure multiparty computation to adjust the class weight for the imbalanced dataset. That is, the privacy-preserving issue of the imbalanced dataset was tackled at the machine-learning algorithm level. The privacy-preserving solution in the machine-learning level is specific. That is, when we change the machine-learning algorithm, a new privacy-preserving solution to the imbalanced data set problem should be proposed. By contrast, as the privacy-preserving solutions in the data level solve the problem in the preprocessing stage, the output of these solutions can be widely used as they are independent of the machine-learning algorithms. So, in this paper, we focus on the privacy-preserving issue of imbalanced data sets at the data level.

Despite the numerous solutions proposed to solve the imbalanced data sets problems, there is almost none of them attempted to resolve the privacy-preserving issue. To the best of our knowledge, Hong et al. invented a secure collaborative machine-learning solution, in which they used a secure multiparty computation to adjust the class weight of the imbalanced dataset. They tackled the privacy-preserving issue of the imbalanced dataset on the machine-learning algorithm level. This solution, however, is sensitive to the algorithm used for machine-learning, i.e., when the machine-learning algorithm is changed, a new privacy-preserving solution must be proposed for the imbalanced dataset problem. In contrast, as the privacy-preserving solutions at the data level work by solving the problem in the preprocessing stage, their output can be used widely, regardless of the machine-learning algorithm adopted in the system. Hence, in this paper, we focus on tackling the privacy-preserving issue of imbalanced datasets on the data level.

Currently, Secure Multiparty Computation (SMC) is one of the most widely used techniques to tackle the privacy-preserving issue. In SMC, multiple parties participate in the game with their individual secure inputs and nobody knows

anything of each other's inputs. When the game ends, according to the game rules, some of the parties will obtain the output. The first SMC solution [26] to the millionaire problem was first presented by Yao. Since then, SMC has been developing rapidly. In 2017, Makri et al. [27] proposed SPDZ, a private image classification with SVM using the SMC framework. Mohassel et al. [28] presented a privacy-preserving machine-learning framework, SecureML, in which the privacy-preserving issue of the linear regression, logistic regression, and neural network training using the stochastic gradient descent method was considered.

In SMC, there are various underlying cryptographic tools, such as garbled circuit, homomorphic encryption scheme, oblivious transfer, and secret sharing scheme. In this paper, we focus on handling the imbalanced dataset problem with the privacy-preserving two-party computation using the homomorphic encryption scheme. Homomorphic encryption is one of the most active research areas in the field of cryptography. Homomorphic encryption was initially proposed by Rivest et al. [29] in 1978. In 1985, ElGamal et al. [30] proposed a widely used multiplicatively homomorphic encryption scheme, known as ElGamal scheme. In 2001, Damgard et al. [31] promoted an additively homomorphic encryption scheme, named Paillier scheme. In 2009, Gentry [32] proposed a fully homomorphic encryption scheme, a ground-breaking development to homomorphic encryption study. Currently, the two most widely used fully homomorphic encryption schemes are the BGV scheme [33] by Brakerski et al. and BFV scheme [34] by Fan et al. In 2021, Chen et al. [35] presented a dynamic multikey fully homomorphic encryption scheme based on LWE assumption in the public key setting.

1.1. Contributions. In this paper, we propose two novel privacy-preserving oversampling protocols, namely PPSMOS and Borderline-PPSMOS. Both PPSMOS and Borderline-PPSMOS are aimed to solve the problem of the imbalanced dataset while preserving the participants' input and output privacies. With the client and the service denoted as Bob and Alice, respectively, the work in this paper can be generally viewed as follows.

- (1) PPSMOS: This algorithm works in a distributed architecture, where Bob inputs no examples at the beginning of the protocol. All the examples, both majority and minority, are provided by Alice. After the protocol, Bob gets the synthetic minority example while he learns nothing of Alice's examples. At the same time, Alice learns nothing of the output Bob receives. PPSMOS shows to be a good solution with a privacy-preserving manner for data balance problems encountered in the cold start phase of many real-life applications.
- (2) Borderline-PPSMOS: In this algorithm, at the start of the protocol, Bob has some majority examples as his input. Meanwhile, Alice has a number of minority examples. After the protocol, Bob receives synthetic minority examples, while he learns nothing of Alice's

minority examples, and Alice learns nothing of Bob's input and output.

- (3) PPSMOS and Borderline-PPSMOS performance analysis: Our analysis shows that PPSMOS generally works more efficiently than Borderline-PPSMOS, while Borderline-PPSMOS achieves a better TP rate and F-Value than PPSMOS. We also found that PPSMOS and Borderline-PPSMOS are both secure in the semihonest model.

1.2. Roadmap of This Paper. The rest of this paper is organized as follows. In Section 2, we introduce the preliminaries. We present the Privacy-Preserving Synthetic Minority Oversampling (PPSMOS) protocol in Section 3 and Borderline-PPSMOS in Section 4. We compare and analyse our protocols in Section 5. We, then, give our concluding remarks in Section 6.

2. Preliminaries

2.1. Homomorphic Encryption. The homomorphic encryption scheme allows us to operate the ciphertext directly. The result obtained after the application of this scheme is equivalent to the ciphertext obtained after performing an operation on a plaintext. Homomorphic encryption algorithms are divided into three categories: additive homomorphism, multiplicative homomorphism, and full homomorphism. For our protocols, we adopt the fully homomorphic encryption scheme. We describe the fully homomorphic encryption algorithm as follows.

We denote (pk, sk) as the system keys, where pk is the public key and sk is the secret key. Furthermore, $E(\alpha)$ is the encryption operation on the plaintext α and $D(\beta)$ is the decryption operation on the ciphertext β . The fully homomorphic encryption scheme follows the properties below.

$$\begin{aligned} E(\alpha) + E(\gamma) &= E(\alpha + \gamma), \\ E(\alpha) * E(\gamma) &= E(\alpha * \gamma). \end{aligned} \quad (1)$$

2.2. Semihonest Model. There are two widely used adversarial models in SMC, the semihonest model, and the malicious model. In this work, we design our protocols in the semihonest model.

In the semihonest model, there are two kinds of participants, the honest participants and the semihonest participants. The honest participants follow the protocol without doing any other activities. At the same time, the semihonest participants followed the protocol and collected the data they obtained during the process of the protocol. After the protocol, they may want to infer information from the data they collected. A protocol is secure in the semihonest model if the semihonest participants get no valuable information from the data they collected.

3. Privacy-Preserving Synthetic Minority Oversampling Protocol

In this section, we present our Privacy-Preserving Synthetic Minority Oversampling Protocol (PPSMOS) and analyze its security aspect.

Suppose that Alice has the total dataset $P = \{p_1, p_2, \dots, p_h\}^T$ with $p_i = (p_i^{(1)}, p_i^{(2)}, \dots, p_i^{(n)})$ with $1 \leq i \leq h$. To simplify, Alice puts all the minority samples in front of P . In other words, we denote the minority subclass by $P_{\min} = \{p_1, p_2, \dots, p_m\}^T$ where m is the number of the minority samples. Both Alice and Bob wish to generate a minority sample p_{new} based on P . After the protocol, Bob gets the output p_{new} under the condition that Alice and Bob cannot know any information about p_{new} and P , respectively.

3.1. PPSMOS

3.1.1. Input. Alice inputs $P = \{p_1, p_2, \dots, p_h\}^T$, where $p_i = (p_i^{(1)}, p_i^{(2)}, \dots, p_i^{(n)})$ and $1 \leq i \leq h$, with the first m elements $P_{\min} = \{p_1, p_2, \dots, p_m\}^T$ belonging to the minority class. Bob inputs nothing.

3.1.2. Output. Bob obtains a newly synthesized minority sample p_{new} while Alice gets nothing.

3.1.3. Preprocessing Stage

- (1) Alice calls the key generation algorithm of the fully homomorphic encryption system to generate the system key (pk, sk) .
- (2) Alice computes the ciphertext $E(P)$ as follows.
 - (i) $E(P) = \{E(p_1), E(p_2), \dots, E(p_h)\}^T$ where $E(p_i) = (E(p_i^{(1)}), E(p_i^{(2)}), \dots, E(p_i^{(n)}))$
- (3) Alice constructs a matrix S that contains the indices of the k -nearest neighbors of every element in P_{\min} , i.e., every s_{ij} in S presents the index of the j^{th} nearest neighbor of p_i in the minority class P_{\min} .

$$S = \begin{pmatrix} s_{11} & \cdots & s_{1k} \\ \vdots & \ddots & \vdots \\ s_{m1} & \cdots & s_{mk} \end{pmatrix}. \quad (2)$$

- (4) Alice discloses $pk, E(P)$ and S on the network.
- (5) Bob gets $pk, E(P)$ and S published by Alice.

3.1.4. Processing Stage

- (1) Bob generates two random integers, α and β , where $1 \leq \alpha \leq m$ and $1 \leq \beta \leq k$.
- (2) Bob generates two random numbers gap and noise where $0 < \text{gap} < 1$. Then, using the public key pk and the encryption algorithm $E(*)$, he computes the ciphertext $E(\text{gap})$ and $E(\text{noise})$.

- (3) Using both ciphertexts obtained in (2), Bob does the following operation to produce X . Then he sends X to Alice.

$$\begin{aligned}
X &= \left[E(\text{gap}) * \left(E(p_{s_{\alpha\beta}}) - E(p_\alpha) \right) + E(p_\alpha) \right] * E(\text{noise}) \\
&= \left[\left(E(\text{gap}) * \left(E(p_{s_{\alpha\beta}}^{(1)}) - E(p_\alpha^{(1)}) \right) + E(p_\alpha^{(1)}) \right) * E(\text{noise}) \right. \\
&\quad \left. \left(E(\text{gap}) * \left(E(p_{s_{\alpha\beta}}^{(2)}) - E(p_\alpha^{(2)}) \right) + E(p_\alpha^{(2)}) \right) * E(\text{noise}) \right. \\
&\quad \dots \\
&\quad \left. \left(E(\text{gap}) * \left(E(p_{s_{\alpha\beta}}^{(n)}) - E(p_\alpha^{(n)}) \right) + E(p_\alpha^{(n)}) \right) * E(\text{noise}) \right]^T.
\end{aligned} \tag{3}$$

- (4) Alice decrypts X using the secret key sk and obtains $D(X) = (\gamma^{(1)}, \gamma^{(2)}, \dots, \gamma^{(n)})$, before sending it to Bob.
- (5) Bob gets the final result p_{new} as follows.

$$p_{\text{new}} = \left(\frac{\gamma^{(1)}}{\text{noise}}, \frac{\gamma^{(2)}}{\text{noise}}, \dots, \frac{\gamma^{(n)}}{\text{noise}} \right). \tag{4}$$

3.2. Security Analysis

Theorem 1. *Under the assumption that the underlying fully homomorphic encryption scheme is secure, PPSMOS securely generates the minority samples in the semihonest model.*

Proof. First, we analyse the situation where Alice is corrupted. In PPSMOS, Alice receives X from Bob. Using the secret key, Alice is able to recover the plaintext:

$$D(X) = (\gamma^{(1)}, \gamma^{(2)}, \dots, \gamma^{(n)}) = \left[\text{gap} * \left(p_{s_{\alpha\beta}} - p_\alpha \right) + p_\alpha \right] * \text{noise}. \tag{5}$$

As α , β , and gap are random numbers, Alice does not have the ability to infer the matchup between $D(X)$ and its samples. Furthermore, since $D(X)$ are confused by the random number noise, Alice has no way of knowing Bob's newly generated point p_{new} . Hence, even if Alice is corrupted, Bob's output is isolated from Alice and, thus, secure.

Next, we analyze the case that Bob is corrupted. In the preprocessing stage, Bob gets the ciphertext $E(P)$ and a matrix S , which are both disclosed by Alice. As the underlying homomorphic encryption scheme is secure in the semihonest model, Bob will not be able to infer any information regarding Alice's private input from $E(P)$. As S presents the index of the j^{th} nearest neighbor of p_i in the minority class P_{min} , Bob is unable to get any information of the specific point of P through S . Therefore, even if Bob is corrupted, Alice's private information is still secure and undisclosed from Bob.

Thus, we can deduct that that Theorem 1 holds. \square

4. Borderline Privacy-Preserving Synthetic Minority Oversampling Protocol

In this section, we present our Borderline Privacy-Preserving Synthetic Minority Oversampling Protocol (Borderline-PPSMOS) and analyze its security aspect.

Suppose that Alice has a minority class $P = \{p_1, p_2, \dots, p_m\}^T$, where $p_i = (p_i^{(1)}, p_i^{(2)}, \dots, p_i^{(n)})$. Bob has a majority class $Q = \{q_1, q_2, \dots, q_t\}^T$, where $q_i = (q_i^{(1)}, q_i^{(2)}, \dots, q_i^{(n)})$. Both Alice and Bob wish to generate a minority sample p_{new} based on P and Q . After the protocol, Bob gets the output p_{new} . Meanwhile, Alice cannot know any information about p_{new} and Q , and Bob cannot know any information about P .

4.1. Borderline-PPSMOS

4.1.1. Input. Alice inputs $P = \{p_1, p_2, \dots, p_m\}^T$ where $p_i = (p_i^{(1)}, p_i^{(2)}, \dots, p_i^{(n)})$. Bob inputs $Q = \{q_1, q_2, \dots, q_t\}^T$ where $q_i = (q_i^{(1)}, q_i^{(2)}, \dots, q_i^{(n)})$.

4.1.2. Output. Alice gets nothing. Bob obtains a newly synthesized minority sample p_{new} .

4.1.3. Preprocessing Stage

- (1) Alice generates the key $s(\text{pk}, \text{sk})$ of the fully homomorphic encryption system.
- (2) Alice computes the ciphertext $E(P)$ as follows.
 - (i) $E(P) = \{E(p_1), E(p_2), \dots, E(p_m)\}^T$ where $E(p_i) = (E(p_i^{(1)}), E(p_i^{(2)}), \dots, E(p_i^{(n)}))$
- (3) Alice constructs a matrix D , where d_{ij} in D represents the square power of the Euclidean distance between the point p_i and its j^{th} -nearest neighbors.

$$D = \begin{pmatrix} d_{11} & \dots & d_{1k} \\ \vdots & \ddots & \vdots \\ d_{m1} & \dots & d_{mk} \end{pmatrix}. \tag{6}$$

- (4) Alice encrypts every element in D and obtains $E(D)$.
- (5) Alice discloses $\text{pk}, E(P)$ and $E(D)$ on the network.
- (6) Bob gets $\text{pk}, E(P)$ and $E(D)$ which were published by Alice.

4.1.4. Processing Stage

- (1) Bob computes $E(Q)$ using pk and the encryption algorithm $E(*)$.
 - (i) $E(Q) = \{E(q_1), E(q_2), \dots, E(q_t)\}^T$, where $E(q_i) = (E(q_i^{(1)}), E(q_i^{(2)}), \dots, E(q_i^{(n)}))$
- (2) For every element p_i in P , Bob calculates the ciphertext of the square power of the Euclidean distance between the p_i and the elements in Q .

$$E(V) = \begin{pmatrix} E(v_{11}) & \cdots & E(v_{1t}) \\ \vdots & \ddots & \vdots \\ E(v_{m1}) & \cdots & E(v_{mt}) \end{pmatrix}. \quad (7)$$

(i) where $E(v_{ij}) = (E(p_i^{(1)}) - E(q_j^{(1)}))^2 + (E(p_i^{(2)}) - E(q_j^{(2)}))^2 + \cdots + (E(p_i^{(n)}) - E(q_j^{(n)}))^2$

$$E(G) = \begin{pmatrix} E(d_{11}) + E(\sigma_1) & \cdots & E(d_{1k}) + E(\sigma_1) & E(v_{11}) + E(\sigma_1) & \cdots & E(v_{1t}) + E(\sigma_1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ E(d_{m1}) + E(\sigma_m) & \cdots & E(d_{mk}) + E(\sigma_m) & E(v_{m1}) + E(\sigma_m) & \cdots & E(v_{mt}) + E(\sigma_m) \end{pmatrix}. \quad (8)$$

(5) Bob performs row and column confusion on $E(G)$ to obtain the confused matrix $E(G')$. Then he sends $E(G')$ to Alice.

(6) Alice receives $E(G')$ and decrypts it with the private key sk to obtain the matrix G' .

$$G' = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1(n+t)} \\ \vdots & \vdots & \ddots & \vdots \\ g_{m1} & g_{m2} & \cdots & g_{m(n+t)} \end{pmatrix}. \quad (9)$$

(7) For every row in G' , Alice computes the k -smallest value and denotes the position of these elements in the matrix R . Then, Alice sends R to Bob.

$$R = \begin{pmatrix} r_{11} & \cdots & r_{1k} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mk} \end{pmatrix}. \quad (10)$$

(8) Bob performs the inverse obfuscation on matrix R to get matrix B .

$$B = \begin{pmatrix} b_{11} & \cdots & b_{1k} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mk} \end{pmatrix}. \quad (11)$$

(9) For the i^{th} row in B , where $1 \leq i \leq k$, Bob counts the number cnt_i of the elements smaller than $k + 1$. Similar to that in Borderline-SMOTE, we call the point $p_i \in \text{Danger}$, if $(k/2) \leq cnt_i < k$.

(10) Bob randomly selects a point p_i from the class Danger and randomly selects an element b_{ix} , which is greater than k from the i^{th} row of B .

(11) Bob generates two random numbers, gap and $noise$, where $0 < gap < 1$. Next, he generates the ciphertext $E(gap)$ and $E(noise)$ by using the public key pk and the encryption algorithm $E(*)$.

(12) Bob performs an operation using the ciphertext $E(gap)$ and $E(noise)$ to obtain X as follows. Then he sends X to Alice.

(3) Bob generates m random numbers $\sigma_1, \sigma_2, \dots, \sigma_m$. Then, he obtains the ciphertext $E(\sigma_1), E(\sigma_2), \dots, E(\sigma_m)$, using the public key pk and the encryption algorithm $E(*)$.

(4) Bob connects $E(V)$ to the encryption matrix $E(D)$ to form a new matrix $E(G)$.

$$\begin{aligned} X &= [E(gap) * (E(q_{b_{ix}}) - E(p_i)) + E(p_i)] * E(noise) \\ &= [(E(gap) * (E(q_{b_{ix}}^{(1)}) - E(p_i^{(1)})) + E(p_i^{(1)})) * E(noise) \\ &\quad (E(gap) * (E(q_{b_{ix}}^{(2)}) - E(p_i^{(2)})) + E(p_i^{(2)})) * E(noise), \dots, \\ &\quad (E(gap) * [E(q_{b_{ix}}^n) - E(p_i^n)] + E(p_i^n)) * E(noise)]^T. \end{aligned} \quad (12)$$

(13) Alice decrypts X using the secret key sk and obtains $D(X) = (\gamma^{(1)}, \gamma^{(2)}, \dots, \gamma^{(n)})$. She then proceeds to send $D(X)$ to Bob.

(14) Bob gets the final result p_{new} as follows.

$$p_{\text{new}} = \left(\frac{\gamma^{(1)}}{\text{noise}}, \frac{\gamma^{(2)}}{\text{noise}}, \dots, \frac{\gamma^{(n)}}{\text{noise}} \right). \quad (13)$$

4.2. Security Analysis

Theorem 2. Under the assumption that the underlying fully homomorphic encryption scheme is secure, Borderline-PSPMOS securely generates the minority sample in the semihonest model.

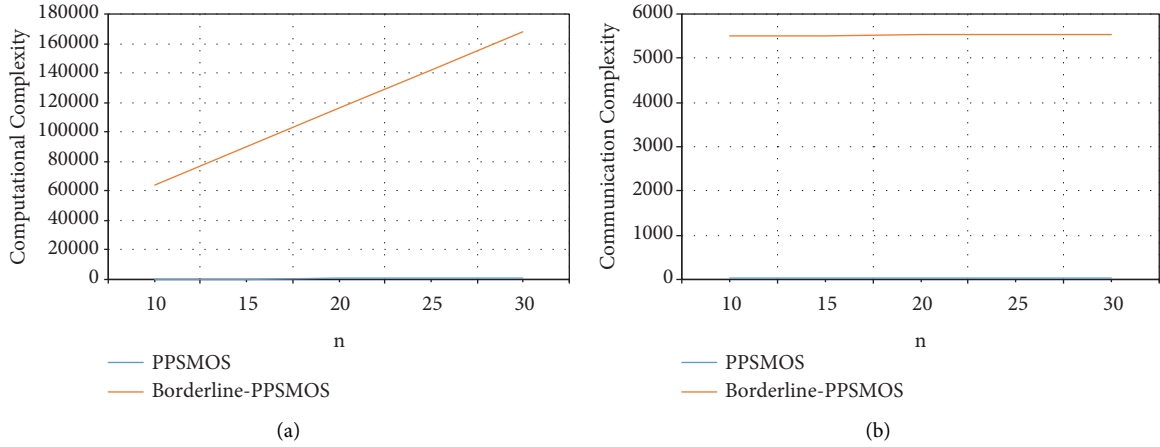
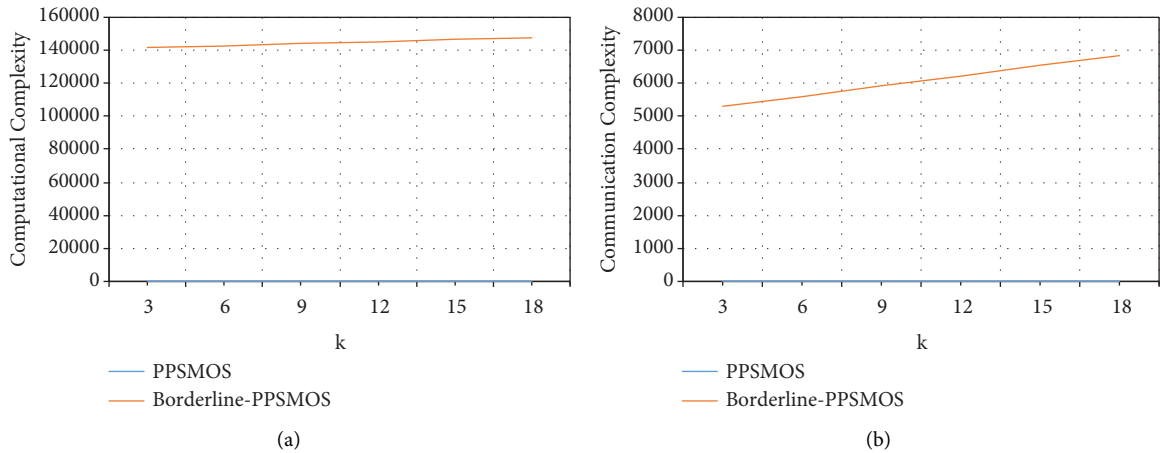
Proof. First, we analyse the situation where Alice is corrupted. In Borderline-PSPMOS, Alice receives $E(G')$ from Bob. Alice is able to recover the plaintext using the private key. However, since Bob obtained $E(G')$ by applying row and column confusion operation on $E(G)$, Alice will not be able to infer the true rank order of $E(G)$. Furthermore, as $E(G')$ is confused by using $E(\sigma_1), E(\sigma_2), \dots, E(\sigma_m)$, where $\sigma_1, \sigma_2, \dots, \sigma_m$ are random numbers, Alice will not be able to know the information of set Q owned by Bob.

Also, when Alice receives X from Bob, she can recover the plaintext:

$$\begin{aligned} D(X) &= (\gamma^{(1)}, \gamma^{(2)}, \dots, \gamma^{(n)}) \\ &= [gap * (q_{b_{ix}} - p_i) + p_i] * \text{noise}. \end{aligned} \quad (14)$$

TABLE 1: Performance analysis of PPSMOS and Borderline-PPSMOS.

Protocol	Preprocessing stage		Processing stage	
	Computational complexity	Communication complexity	Computational complexity	Communication complexity
PPSMOS	hn	hn	$5n + 2$	n
Borderline-PPSMOS	$mn + mk$	$tn + mk$	$(m + mt + 2t + 3)n + 4mk + 2mt + m + 2$	$m(k + t) + n$

FIGURE 1: Computational Complexity (left) and Communication Complexity (right) of PPSMOS and Borderline-PPSMOS by varying n when $m = 100, t = 50, k = 5$.FIGURE 2: Computational Complexity (left) and Communication Complexity (right) of PPSMOS and Borderline-PPSMOS by varying k when $m = 100, t = 50, n = 25$.

However, as i, x and gap are random numbers, Alice does not have the ability to infer the matchup between $D(X)$ and its samples. In addition, as $D(X)$ is confused by the random number noise, Alice has no way of knowing Bob's newly generated point p_{new} . Hence, even if Alice is corrupted, Bob's input and output are totally isolated from Alice and still secure.

Next, we analyze the case where Bob is corrupted. In the preprocessing stage, Bob gets the ciphertext $E(P)$, $E(D)$ and public key pk . As the underlying homomorphic encryption scheme is secure in the semihonest model, Bob is unable to

infer any information regarding Alice's private input from $E(P)$ and $E(D)$. In the processing stage, Alice computes the k -smallest value and denotes the position of these elements in matrix R . During this step, Alice only sends the location index to Bob, which does not reveal any information of Alice's input. Next, Bob gets $D(X) = (\gamma^{(1)}, \gamma^{(2)}, \dots, \gamma^{(n)})$ from Alice. Similarly, as the homomorphic encryption scheme is secure in the semihonest model, Bob cannot infer p_i from $\text{gap} * (q_{b_{ix}} - p_i) + p_i$. Thus, even if Bob is corrupted, Alice's private information is still secure and undisclosed from Bob.

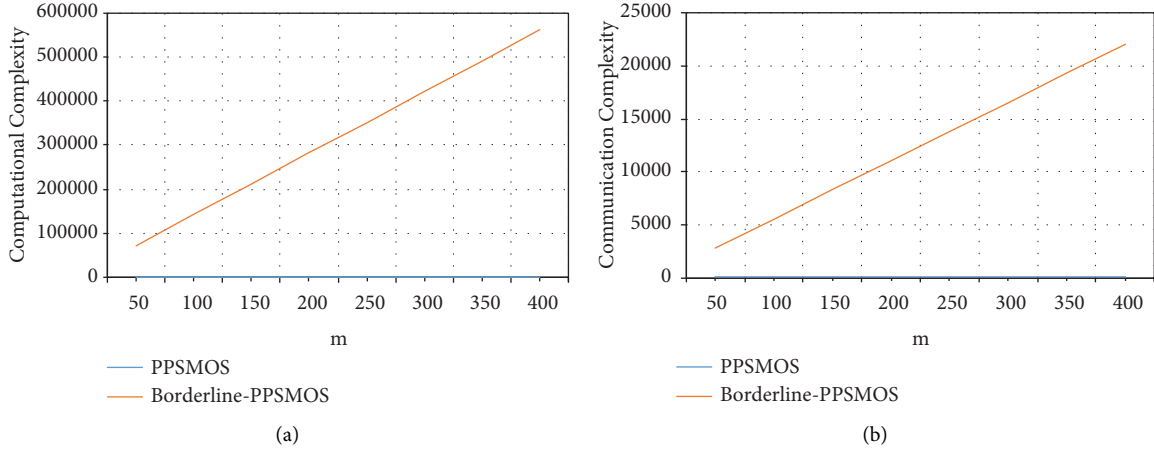


FIGURE 3: Computational Complexity (left) and Communication Complexity (right) of PPSMOS and Borderline-PPSMOS by varying m when $n = 25, t = 50, k = 5$.

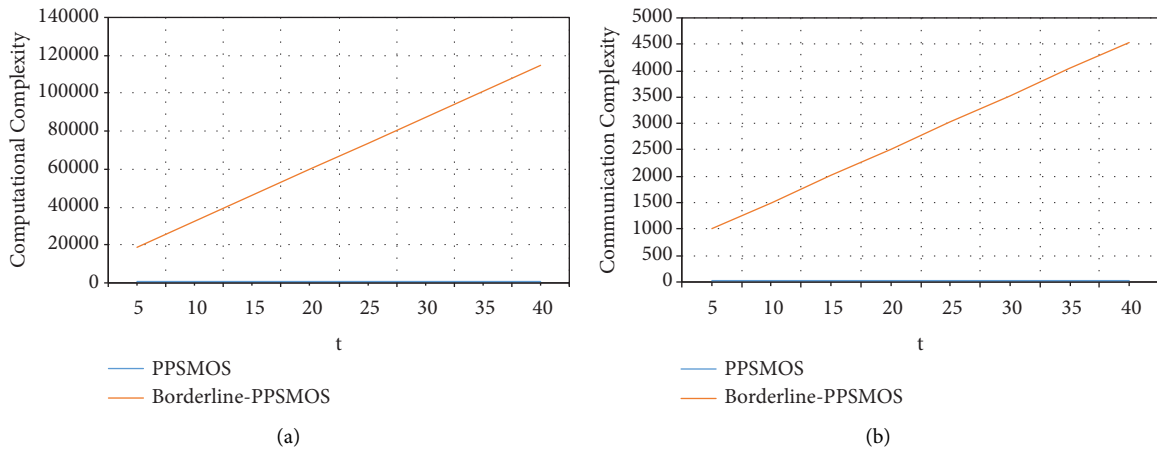


FIGURE 4: Computational Complexity (left) and Communication Complexity (right) of PPSMOS and Borderline-PPSMOS for the cost by varying t when $m = 100, n = 25, k = 5$.

Therefore, we can deduce that Borderline-PPSMOS is secure in the semihonest model, under the fully homomorphic encryption scheme, i.e., Theorem 2 holds. \square

5. Performance Analysis

In this section, we present the performance analysis of both PPSMOS and Borderline-PPSMOS. On the efficiency analysis, we look at computational complexity and communication complexity. Given that h is the size of Alice's input in PPSMOS, n is the feature size, t is the size of the majority class, m is the size of the minority class, and k is a parameter, we analyze the protocols' performances as follows.

First, we analyze the performance of PPSMOS. During the preprocessing stage, Alice performs the encryption operation for hn times while Bob gets hn ciphertexts. Furthermore, in the processing stage, Bob performs encryption operation 2 times and $2n$ times for each homomorphic additive operation and homomorphic multiplicative

operation. Alice, then, performs decryption operations for n times. Bob sends n ciphertexts to Alice.

Secondly, we analyze the efficiency of Borderline-PPSMOS. In the preprocessing stage, Alice performs the encryption operations for $mn + mk$ times, while Bob gets $tn + mk$ ciphertexts. Next, in the processing stage, Bob performs encryption operation for $tn + m + 2$ times, homomorphic additive operation for $mtn + (t + k)m + 2n$ times, and homomorphic multiplicative operation for $mtn + 2n$ times. Alice performs decryption operations for $m(k + t) + n$ times. Finally, Bob transferred $m(k + t) + n$ ciphertexts to Alice.

We summarise the computational complexity and communication complexity of both protocols below in Table 1.

We visualize the operational efficiency of both PPSMOS and Borderline-PPSMOS during the processing stage by instantiating the parameters, as shown below in Figures 1–4.

From these figures, we can conclude the following: (1) the computational complexity and communication complexity of

PPSMOS are lower than those of Borderline-PPSMOS; (2) the computational complexity and communication complexity of PPSMOS depend only on the feature size n ; (3) the computational complexity and communication complexity of Borderline-PPSMOS depend on almost all of the parameters, i.e., n , t , m and k ; (4) Borderline Synthetic Minority Oversampling Protocol achieves better TP rate and F-Value than Synthetic Minority Oversampling Protocol [22]. Furthermore, as our privacy-preserving schemes do not affect the TP rate and F-Value of the underlying Minority Oversampling protocol, we can further deduct that Borderline-PPSMOS achieves a better TP rate and F-Value than PPSMOS.

6. Conclusion

In this paper, we propose two novel privacy-preserving oversampling protocols, PPSMOS and Borderline-PPSMOS, that are aimed to address the imbalanced dataset issue while preserving the privacy of the participants' input and output.

PPSMOS works in a manner where the client inputs no majority examples, as opposed to Borderline-PPSMOS, where the client has some majority examples. Both PPSMOS and Borderline-PPSMOS are secure in the semihonest model. This means that both methods are suitable for the preprocessing stage of machine-learning and applicable to any cases where synthesizing minority examples in a privacy-preserving manner is needed. Our results show that PPSMOS is more efficient than Borderline-PPSMOS in general, while Borderline-PPSMOS achieves better TP rate and F-Value than PPSMOS.

While doing our work in the semihonest model and through our analysis, we found that our protocols are unable to resist malicious attacks, and their efficiency needs improvements. As future work, we will continue improving our research on these two aspects, as well as focusing on designing better privacy-preserving protocols that are to be used in the preprocessing stage of machine-learning.

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 there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by Project of High-level Teachers in Beijing Municipal Universities in the Period of 13th Five-year Plan (CIT&TCD201904097) and The Fundamental Research Funds for Beijing Local Universities From Capital University of Economics and Business.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

- [2] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Proceedings of the International conference on financial cryptography and data security*, pp. 436–454, Christ Church, Barbados, May 2014.
- [3] C. Hou, M. Zhou, Y. Ji et al., "SquirrelL: Automating attack analysis on blockchain incentive mechanisms with deep reinforcement learning," arXiv preprint arXiv:1912.01798, 2019.
- [4] T. Li, Y. Chen, Y. Wang et al., "Rational protocols and Attacks in blockchain system," *Security and Communication Networks*, vol. 2020, no. 44, 11 pages, Article ID 8839047, 2020.
- [5] K. Suzuki, "Pixel-based machine learning in medical imaging," *International Journal of Biomedical Imaging*, vol. 2012, Article ID 792079, 1 page, 2012.
- [6] M. L. Giger and K. Suzuki, "Computer-aided diagnosis (CAD)," *Biomedical Information Technology*, vol. 31, pp. 359–374, 2007.
- [7] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "PSSPR: A source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1204–1221, 2022.
- [8] K. Doi, "Current status and future potential of computer-aided diagnosis in medical imaging," *British Journal of Radiology*, vol. 78, no. 1, pp. S3–S19, 2005.
- [9] S. M. Usman, M. Usman, and S. Fong, "Epileptic seizures prediction using machine learning methods," *Computational and Mathematical Methods in Medicine*, vol. 2017, Article ID 9074759, 2017.
- [10] S. R. Girard, V. Legault, G. Bois, and J. F. Boland, "Avionics graphics hardware performance prediction with machine learning," *Scientific Programming*, vol. 2019, Article ID 9195845, 15 pages, 2019.
- [11] T. Ali, H. Khazaei, M. H. Y. Moghaddam, and Y. Hassan, "Machine learning in transportation," *Journal of Advanced Transportation*, vol. 2019, Article ID 4359785, 3 pages, 2019.
- [12] Y. Chen, K. Liu, Y. Xie, and M. Hu, "Financial trading strategy system based on machine learning," *Mathematical Problems in Engineering*, vol. 2020, Article ID 3589198, 13 pages, 2020.
- [13] J. Mu, F. Wu, and A. Zhang, "Housing value forecasting based on machine learning methods," *Abstract and Applied Analysis*, vol. 2014, Article ID 648047, 7 pages, 2014.
- [14] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, pp. 1–22, 2021.
- [15] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
- [16] J. Zhang, L. Chen, and F. Abid, "Prediction of Breast cancer from imbalance respect using cluster-based undersampling method," *Journal of Healthcare Engineering*, vol. 2019, Article ID 7294582, 10 pages, 2019.
- [17] H. Wang and X. Liu, "Undersampling bankruptcy prediction: taiwan bankruptcy data," *PLoS ONE*, vol. 16, no. 7, pp. 1–17, 2021.
- [18] H.-L. Dai, "Class imbalance learning via a fuzzy total margin based support vector machine," *Applied Soft Computing*, vol. 31, pp. 172–184, 2015.
- [19] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no. 3, pp. 408–421, 1972.
- [20] Z. Wang, C. Cao, and Y. Zhu, "Entropy and confidence-based undersampling boosting random forests for imbalanced problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5178–5191, 2020.

- [21] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [22] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proceedings of the International conference on intelligent computing*, Springer, Berlin, Heidelberg, pp. 878–887, 2005.
- [23] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," *Information Sciences*, vol. 465, pp. 1–20, 2018.
- [24] L. Li, H. He, and J. Li, "Entropy-based sampling Approaches for multi-class imbalanced problems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 11, pp. 2159–2170, 2020.
- [25] C. Hong, Z. Huang, W. J. Lu et al., "Privacy-preserving collaborative machine learning on genomic data using TensorFlow," in *Proceedings of the ACM Turing Celebration Conference-China*, pp. 39–44, Hefei, China, May 2020.
- [26] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pp. 160–164, Chicago, IL, USA, November 1982.
- [27] E. Makri, D. Rotaru, N. P. Smart, and F. Vercauteren, "EPIC: efficient private image classification (or: Learning from the masters)," *Cryptographers' Track at the RSA Conference, Lecture Notes in Computer Science*, vol. 11405, pp. 473–492, 2019.
- [28] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *Proceedings of the 2017 IEEE symposium on security and privacy*, pp. 19–38, San Jose, CA, USA, May 2017.
- [29] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data Banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, pp. 120–126, 1978.
- [30] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *International Cryptology Conference*, vol. 31, no. 4, pp. 10–18, 1985.
- [31] I. Damgård and M. Jurik, "A generalisation, a simplification and some Applications of paillier's probabilistic public-key system," *Public Key Cryptography*, Springer, in *Proceedings of the International workshop on public key cryptography*, pp. 119–136, Berlin, Heidelberg, 2001.
- [32] C. Gentry and D. Boneh, "A fully homomorphic encryption scheme," *Stanford university*, vol. 20, no. 9, 2009.
- [33] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory*, vol. 6, no. 3, pp. 1–36, 2014.
- [34] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *Cryptology ePrint Archive*, p. 144, 2012.
- [35] Y. Chen, S. Dong, T. Li, Y. Wang, and H. Zhou, "Dynamic multikey FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5239–5249, 2021.

Research Article

Privacy-Preserving Collaborative Computation for Human Activity Recognition

Lin Wang,^{1,2} Chuan Zhao ,^{1,2,3} Kun Zhao ,⁴ Bo Zhang ,^{1,2} Shan Jing,^{1,2} Zhenxiang Chen,^{1,2} and Kuiheng Sun^{1,2}

¹School of Information Science and Engineering, University of Jinan, Jinan 250022, China

²Shandong Provincial Key Laboratory of Network-Based Intelligent Computing, University of Jinan, Jinan 250022, China

³Shandong Provincial Key Laboratory of Software Engineering, Jinan, China

⁴Inspur Electronic Information Industry Co. Ltd., Beijing, China

Correspondence should be addressed to Chuan Zhao; ise_zhaoc@ujn.edu.cn and Kun Zhao; zhaokunbj@inspur.com

Received 17 November 2021; Accepted 28 January 2022; Published 28 February 2022

Academic Editor: Yuling Chen

Copyright © 2022 Lin Wang 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.

Human Activity Recognition (HAR) enables computer systems to assist users with their tasks and improve their quality of life in rehabilitation, daily life tracking, fitness, and cognitive disorder therapy. It is a hot topic in the field of machine learning, and HAR is gaining more attention among researchers due to its unique societal and economic advantages. This paper focuses on a collaborative computation scenario where a group of participants will securely and collaboratively train an accurate HAR model. The training process requires collecting a massive number of personal activity features and labels, which raises privacy problems. We decentralize the training process locally to each client in order to ensure the privacy of training data. Furthermore, we use an advanced secure aggregation algorithm to ensure that malicious participants cannot extract private information from the updated parameters even during the aggregation phase. Edge computing nodes have been introduced into our system to address the problem of data generation devices' insufficient computing power. We replace the traditional central server with smart contract to make the system more robust and secure. We achieve the verifiability of the packaged nodes using the publicly auditability feature of blockchain. According to the experimental data, the accuracy of the HAR model trained by our proposed framework reaches 93.24%, which meets the applicability requirements. The use of secure multiparty computation techniques unavoidably increases training time, and experimental results show that a round of iterations takes 36.4 seconds to execute, which is still acceptable.

1. Introduction

Human Activity Recognition (HAR) is a machine learning task to identify human activities through images, videos, or sensor data generated by smart wearable devices. HAR has a wide range of applications today, such as monitoring the health of individuals by recognizing their activities, or it can be used in public places such as plazas and train stations to identify unusual acts of terror in order to give an advance warning [1].

However, when using this type of data, companies or data owners often face a number of issues:

(1) Data owners are reluctant to reveal their information, whether directly or by computational inference

- (2) The massive amount of data generated by IoT devices poses a huge challenge to the storage and processing capacity of central servers
- (3) The network bandwidth cannot handle such an order of magnitude of data transfer
- (4) The node performing the computation can be hijacked or corrupted by adversaries to perform incorrect computation operations

To address the above-mentioned issues, researchers have conducted many explorations. In order to protect privacy and save bandwidth, federated learning is proposed [2]. Federated learning enables us to keep model training procedure on local devices without transmitting data to central

server. At present, federated learning has a mature application in the industrial field [3]. However, with in-depth research, it is found that there are still some problems in federated learning. For example, although federated learning only transmits model update parameters, the updated parameters will still disclose sensitive information to the third party or the central server [4, 5]. Commonly used methods, including secure multiparty computation and differential privacy, aim to resist privacy disclosure in the learning process [4, 6]. However, these approaches are often accompanied by a loss of model efficiency or an increase in training time. Blockchain also has many studies that combine it with federated learning due to its decentralized nature. Kumar et al. use blockchain to first validate the data and then use federated learning to train a deep learning model globally to improve recognition rates against CT images of COVID-19 patients [7]. Qi et al. use a blockchain-based federated learning framework for predicting traffic flow, the model will be verified by miners, the noise will be added to the model to enhance privacy safeguards, and the scheme can effectively prevent poisoning attacks, but there will be some sacrifice in model effectiveness [8]. Edge computing is also commonly used in cutting-edge research in machine learning, where the use of edge nodes to offload computational and storage tasks from a central server can effectively improve training efficiency. Khelifi et al. explored the applicability of deep learning models (i.e., convolutional neural networks, recurrent neural networks, and augmented learning) with IoT devices. The study sought to assess the future trends of deep learning plus edge computing in the future. The study points out that convolutional neural models can be used in the IoT domain and that reliable machine learning models can be trained even with data from complex environments [9]. Secure multiparty computing often plays an important role in this as well. Sangaiah et al. proposed an approach using edge computing plus machine learning to protect the confidentiality of certain location-based services. The approach uses Hidden Markov Models by combining decision trees and k-means algorithms. The benefits of the mobile edge service strategy are location confidentiality and low latency. Both network and computing services are located near the user as a way to achieve lower latency [10, 11].

In this paper, we adopt the idea of federated learning, where users train models locally and optimize the model jointly by uploading parameters instead of uploading data to a central server [2]. We also use a secure aggregation algorithm to eliminate the possibility of the server inferring information via gradients [12, 13]. Furthermore, we consider edge computing and blockchain in our framework. To be specific, we replace the traditional central server with a blockchain. The properties of blockchain make our proposed framework possess a series of security features such as transparency, auditable, and tamper-proof [14]. Edge nodes are introduced in our scheme to relieve the computational pressure and bandwidth pressure on the system [15]. The specific framework structure and the implementation will be presented in Section 3.

In general, our contributions can be summarized as follows:

- (1) We consider federated learning and edge computing scenario to keep private data local instead of being uploaded to the central server, which helps to protect users' privacy. By doing so, we also achieved alleviating the load of the central server, making the computation tasks be processed faster.
- (2) We implement an advanced secure aggregation algorithm that aggregates exactly the results we want, the same as the computed result under plaintext. Also, the whole aggregation and transmission process is in the form of shares, which ensures that the adversary cannot steal information by observing these intermediate shares.
- (3) We deploy smart contracts to replace the traditional central server, avoiding the occurrence of a single point failure of the central server. Moreover, the public auditability feature of blockchain also allows other nodes to verify the aggregation results, thus preventing dishonest behaviors of aggregation nodes.

2. Preliminaries

In this section, we briefly introduce basic tools and corresponding techniques needed in this paper.

2.1. Edge Computing. Edge computing is a distributed computing architecture that refers to distributing computation and storage tasks to edge nodes that are logically closer to users and data sources for processing. This architecture can effectively reduce network latency caused by data transmission, significantly improve the response time of network services, and enhance data security for a better user experience.

In the 1990s, to improve network quality, a research group at MIT proposed CDN (Content Delivery Network) to enable network sites close to users to acquire and cache network content and reduce the footprint on users' broadband. This architecture is widely used in various Internet scenarios [15]. On the other hand, cloud computing was created to cope with the increasing amount of data and computing. The rapid growth and evolution of cloud computing have led to dramatic changes in the way society works and business models [16], but along with development, cloud computing has also revealed many drawbacks. For example, the increasing volume of computation and data not only increases the computational burden on servers but also increases the bandwidth burden on cloud computing centers. This may prolong data processing time and reduce data processing speed and transmission speed. This is fatal for applications such as the Internet of Things, which has a huge amount of data and is latency-sensitive [17].

Edge computing can be seen as a combination of CDN and cloud computing. Due to the advancement of

technology, the performance of devices as edge nodes is also improving, which enables the tasks of edge nodes to be no longer limited to storage but also includes data processing and computing operations such as machine learning. With the development of IoT, edge computing is widely used to process IoT data, which makes edge computing technology have a broader development prospect. The usual architecture of edge computing is shown in Figure 1.

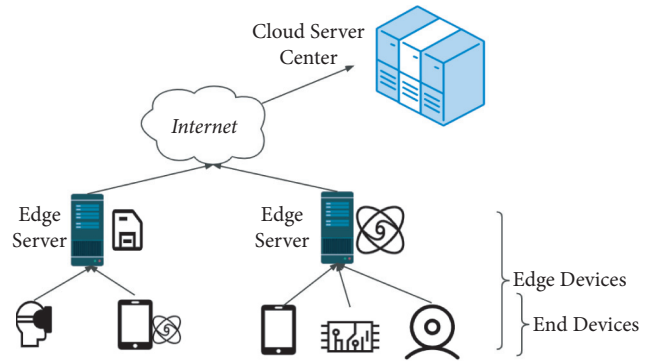


FIGURE 1: Edge computing.

2.2. *Secret Sharing*. Secret sharing refers to schemes for distributing a secret among a group of participants, each of whom is allocated a share of the secret. The secret can be reconstructed only when a sufficient number of shares are combined. Individual shares are of no use on their own. In this paper, we use Shamir’s Secret Sharing, which is formulated by Adi Shamir [12]. Shamir’s Secret Sharing is an ideal and perfect (t, n) -threshold scheme. In such a scheme, the aim is to divide a secret s into n pieces of data s_1, \dots, s_n (known as shares) in such a way that

- (1) Knowledge of any t or more s_i pieces makes s easily computable. That is, the complete secret s can be reconstructed from any combination of t pieces of data
- (2) Knowledge of any $t - 1$ or fewer s_i pieces leaves s completely undetermined, in the sense that the possible values for s seem as likely as with knowledge of 0 pieces. The secret s cannot be reconstructed with fewer than t pieces.

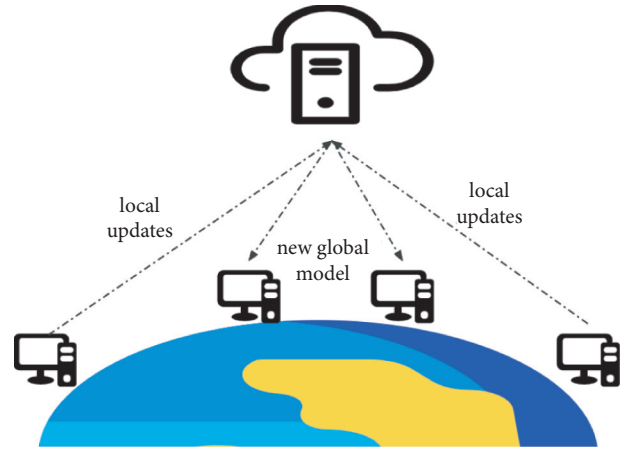


FIGURE 2: Federated learning.

2.3. *Federated Learning*. Federated learning is a distributed machine learning framework proposed by Google that allows multiple users to collaborate on training a global model while maintaining user’s privacy, as shown in Figure 2[18]. In recent years, various countries have established legal restrictions to preserve the privacy of personal information, which makes direct access to user data for machine learning training impossible. Google proposes federated learning, which trains data locally rather than uploading data to a traditional central server to address this issue. This distributed architecture ensures the confidentiality of user data while also optimizing the usage of computing resources on local devices. The central server is only responsible for the coordination, which decreases the server’s processing load. However, some issues must be addressed before federated learning can be used in practice, such as communication issues with a large number of participating devices and system compatibility issues caused by the diversity of participating devices [19]. However, with the increased emphasis on privacy protection, federated learning has become a very promising technology [20].

Definition 1 (federated learning).

Federated learning refers to training a global model using data stored in millions of remote devices, a task that can be represented by the following objective function:

$$\min F(w), \text{ where } F(w) := \sum_{k=1}^m p_k F_k(w), \quad (1)$$

where m represents the total number of devices, F_k is the local objective function of the k -th device, p_k is defined as the influence weight of the corresponding device, p_k has the following properties, $p_k \geq 0$, and $\sum_{k=1}^m p_k = 1$.

2.4. *Smart Contract*. For the first time, Nick Szabo proposed the concept of smart contract in 1995 [21]. Smart contracts are a set of digital contracts that are automatically executed between committed parties. Smart contract is more secure and has lower transaction costs than regular contract. However, due to technological limitations, smart contract could not be executed until the practical implementation of blockchain technology. Blockchain is built on mutually trusted nodes, allowing us fairly and securely to run contracts. There are numerous stable and well-known applications, such as Ether on the public chain and Fabric and Quorum on the nonpublic chain. Smart contract can be developed to extend the functionalities of blockchain beyond digital currency such as Bitcoin, allowing it to be widely used in banking, copyright, and many other industries. However, there are still issues in smart contract that must be addressed, such as unusual programming languages and a lack of

debugging tools, which pose security concerns to smart contract [22].

3. Assumptions and Threats

Our framework is an open machine learning system that allows each node to join or depart at any time. We assume that all nodes want in order to collaborate to train a machine learning model but do not want their data to be utilized or observed by others.

3.1. Design Assumptions

3.1.1. System Topology. We assume that the edge nodes are smart hardware in the home with enough processing power for local training, such as smart gateways, smart routers, or personal computers. Each edge node connects all of the smart devices in the home (e.g., camera, smartphone, and smart-watch). We anticipate that no malicious attacks will be initiated among family members. Thus, we can transmit plaintext between smart devices and edge node without considering encryption. We assume that each edge node can communicate with a subset of other edge nodes, allowing messages to be broadcast from any edge node to all edge nodes.

3.1.2. Machine Learning. We assume that the Genesis Block propagates all training information to all edge nodes. The initial model, hyperparameters, optimization strategies, and learning objectives are all part of this. The edge nodes want to keep the local dataset private during the training phase. We use stochastic gradient descent (SGD) as the optimization algorithm in the local training phase. SGD is a universal optimization technique that may be used to train a wide range of models, including deep neural networks [23].

3.2. Threats. We analyzed possible threats during the training process as follows:

- (1) Users' data can be maliciously analyzed and abused. We must prevent exposing users' plaintext data
- (2) An adversary can deduce user information by seeing updates. User updates should not be directly observed
- (3) Corrupted edge nodes may perform incorrect calculations and submit invalid global models

When data need to be stored on a cloud server, encryption of the data is often an option to prevent the cloud server from stealing the data. However, we assume that an edge node is only responsible for collecting and processing information from family members. Therefore, we do not consider encryption between smart devices and edge devices.

4. Framework Design

4.1. Framework Overview. Our proposed framework's main goals are as follows:

- (1) Data owners collaborate to train an efficient Human Activity Recognition model

- (2) Accelerate the training process by introducing edge computing architecture
- (3) Prevent leakage of user information during the aggregation phase by using secure aggregation algorithms
- (4) Use blockchain public verifiability and tamper-evident to oversee the behavior of packaged nodes

Each node on the blockchain network collects data generated by the smart devices. Each block includes the information generated after one iteration round. Figure 3 shows the process of one iteration.

- (1) *Preparation.* Smart devices collect data on human behaviors using built-in sensors. When certain criteria are satisfied (power and network connection, no other tasks, and sufficient data), smart devices will transmit this data to the associated edge node. Before training begins, all edge nodes on this blockchain network receive an initial random global model from the Genesis Block, used for the first update. This process is shown in Steps 1, 2, and 3 of Figure 3.
- (2) *Local Training.* A local model is calculated using the latest global model and local data. This is Step 4 of Figure 3.
- (3) *Model Aggregation.* With the secret sharing algorithm, each node divides its update into n secret shares (n specifies the number of edge nodes in the distributed ledger) and distributes them to other nodes (Step 5). Step 6 requires all nodes to aggregate the shares they receive and then broadcast the results in Step 7. The first node that receives enough results will reconstruct the global model (Step 8).
- (4) *Submit Block.* Finally, the first node to reconstruct the global model will combine the essential data into a new block and upload this block to the blockchain (Step 9).

4.2. Preparation for Training. Steps 1, 2, and 3 in Figure 3 represent the preparation phase. In this phase, we mainly focus on data collection, data transmission, and creating and distributing Genesis Block.

First, the smart device will collect information about people's activities. When certain conditions are met (e.g., the volume of data is sufficient; power and network are connected), the smart devices will send the data to the associated edge node.

The initial training information will be added to the Genesis Block. We anticipate that a trusted institution will generate the Genesis Block and broadcast it to all edge nodes to begin the training process. The trusted institution is only trusted at this phase, and it will not be involved in the following training process. The Genesis Block provides the model's initial state w_0 and the predicted number of iterations T . There are also public keys P_K for each user i used to generate the commitment to each node's update (detailed in Section 4.5).

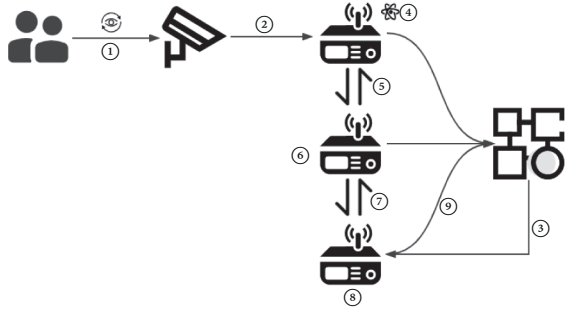


FIGURE 3: Overview.

4.3. Local Training. In the t -th iteration, the global model w_t is downloaded locally from the blockchain by each edge node. Each node has n_k samples, where k is the index of the node. n_k varies among nodes depending on the number of local smart devices and the amount of people's activities on that day. Each edge node computes a local gradient g_k on the current model w_t using its local data n_k . For a given learning rate ε , the local model w_{t+1}^k is given by

$$w_t - \varepsilon g_k \longrightarrow w_{t+1}^k. \quad (2)$$

The hyperparameters required for the computation process, such as the learning rate ε and the client training batch size B_t , are specified by the Genesis Block.

4.4. Aggregation Protocol. Edge nodes use a secret sharing approach to broadcast their local updates to other nodes in the blockchain network. For the following step of verification, they also broadcast the commitment $\text{COMM}(\Delta w_i)_{\text{sign}}$ of their update simultaneously, which carries their signature so that others cannot forge it. The entire aggregation process is described in the following.

The optimization algorithm in our proposed framework is stochastic gradient descent (SGD). Each node computes updates using the latest global model downloaded from blockchain and local data, and all updates are aggregated into a new global model. In the i -th iteration, the following equation is used to update the model parameter w :

$$w_{t+1} = w_t - \eta_t \left(\lambda w_t + \frac{1}{b} \sum_{(x_i, y_i) \in B_t} \Delta l(w_t, x_i, y_i) \right). \quad (3)$$

η_t is the learning rate, λ denotes the regularization parameter, which is used to prevent overfitting, B_t denotes the batch of one training sample of size b , and Δl denotes the gradient of the loss function.

The aggregation protocol requires all edge nodes to collaborate in order to aggregate their local updates into a new global model, and this protocol uses secret sharing to ensure that each node's private data and model updates cannot be seen or inferred by any node other than itself. Algorithm 1 shows the secure aggregation algorithm.

Assume that m edge nodes representing m families collaborate to train a model, and the update i .update for each node i will be encoded as a d -polynomial. This polynomial

will be divided into n secret shares ($n = 2 * (d + 1)$). These n shares are distributed equally among all m nodes, and it takes $(d + 1)$ shares to reconstruct this model, indicating that at least $m/2$ nodes must collaborate to obtain the private data of a specific node. Each edge node i that accumulates enough shares (usually a minimum number u) aggregates those shares and then broadcasts the aggregation_result[i] to all nodes once again. After receiving the aggregated $d + 1$ shares from at least half of the nodes, a node can reconstruct the sum of all local node updates $\sum_{j=1}^u \Delta w_j$. Eventually, the aggregated results of all nodes $\sum_{j=1}^u \Delta w_j$, the latest global model w_t , and all update commitments will be stored in a new block.

4.5. Block Structure. Each block contains a hash pointing to the previous block in order to link to it. Furthermore, malicious edge nodes may perform the aggregation process incorrectly to damage the model. Each block should include a new global model w_t as well as the aggregation results of all node updates $\sum_u \Delta w$ to validate the edge node aggregation procedure. This allows us to test whether the global model is correctly generated by

$$w_t = w_{t-1} + \frac{1}{u} * \left(\sum_u \Delta w \right). \quad (4)$$

Figure 4 shows the blockchain structure that we designed. To ensure that the aggregated results are generated from each node's local update, we keep each node's commitment to their submitted updates in the block as well. Then, the homomorphic nature of the commitment allows us to check if the edge node honestly aggregated the model [24].

$$\text{COMM}\left(\sum \Delta w_i\right) = \prod_i \text{COMM}(\Delta w_i). \quad (5)$$

5. Experiment

We use virtual machines to simulate PCs capable of collecting personal data from smartphones and training local models using the Long Short-Term Memory (LSTM) algorithm. We used three virtual machines for deep learning training, each with 4 GB of RAM and a GTX2080TI GPU. Figure 5 shows the training effect of our proposed framework compared to the training effect of the algorithm using differential privacy. Differential privacy is another prominent strategy in federated learning for protecting personal information. However, using differential privacy often results in decreased accuracy. The results show that our model meets the usability requirement and outperforms the model using differential privacy.

In addition, as shown in Figure 6, we tested the running duration of each component. Because the computation of the security aggregation algorithm is substantially more significant than that of plaintext, a cycle of iteration takes 36.4 seconds, with the process of secure aggregation counting for 72.26% of the overall time. However, this is still acceptable.

```

for each client  $[i] \in m$  do
   $d$  - polynomial  $\leftarrow i$ .update
   $n$  shares  $\leftarrow d$  - polynomial
   $n$  shares are equally distributed among  $m$  nodes
end for
for each client  $[i] \in m$  do
  if client  $[i]$  received  $u$  shares then
    aggregation_results · Share  $[i] \leftarrow$ 
    Aggregate (client_update  $[1]$ .share  $[i], \dots$  client_update  $[u]$ .share  $[i]$ )
    Broadcast the share of aggregation results
  end if
end for
for each client  $[i] \in m$  do
  if client  $[i]$  received  $d + 1$  shares of aggregation results then
    Reconstructing out aggregated results
  end if
end for

```

ALGORITHM 1: Secure aggregation algorithm.

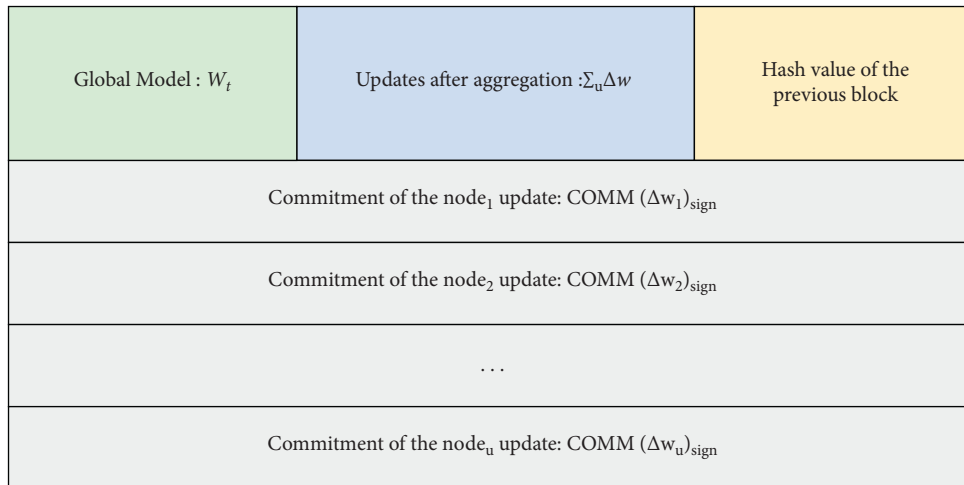


FIGURE 4: Block structure.

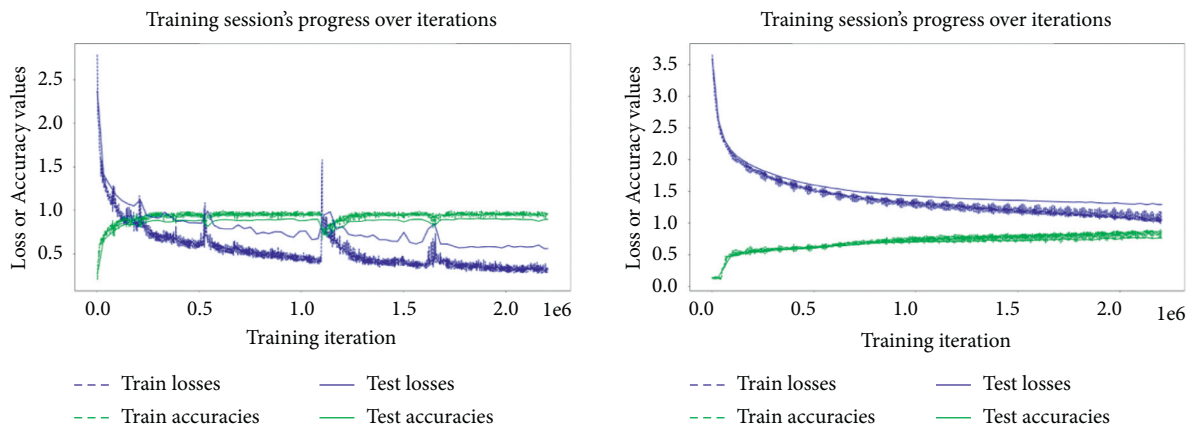


FIGURE 5: Comparison of the two algorithms.

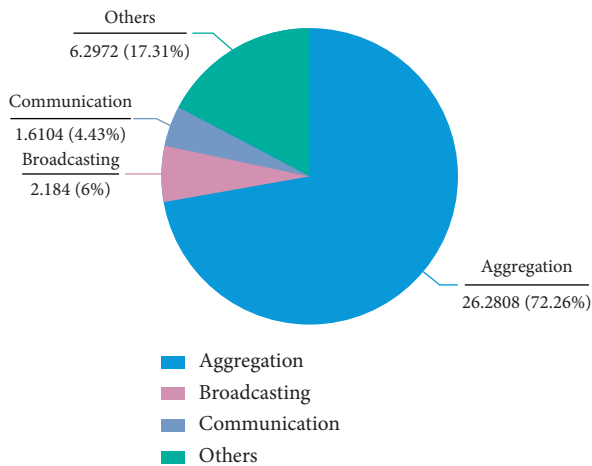


FIGURE 6: Running time of each part.

6. Conclusion

In this paper, we proposed a privacy-preserving collaborative machine learning framework. We combined edge computing architecture with distributed computing to ensure that data are kept local, which ensures that private data are not compromised. In addition, we used a secure aggregation algorithm to ensure that personal information does not leak even throughout the aggregation process. We tested this framework on the HAR dataset and compared the performance of our proposed framework to other popular methods. Our framework can be used for a wide range of different machine learning tasks that require privacy protection.

This framework can be improved in two ways in the future. Firstly, as the number of nodes in the network grows, the effectiveness of our consensus protocol rapidly decreases due to network fluctuations and differences in processing capacity across users. As a result, in the future, we will provide a new consensus mechanism based on consistency hash and proof of stake (PoS). It can also prevent malicious computing nodes from poisoning the model, enhance the efficiency of the consensus process, and reduce energy usage. On the other hand, the edge computing nodes considered in this paper will only cover smart wearable devices from the same family. The edge computing nodes will be home PCs or smart gateways. So, we will overlook data theft and data poisoning at this point. However, we are aware that data and model poisoning attacks can still be carried out between family members. Thus, we will strive to apply anomaly detection methods to identify poisoned data and models in future work.

Data Availability

The dataset can be found on the UCI Machine Learning Repository: Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21st European Symposium on Artificial Neural Networks, Computational Intelligence, and

Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61702218 and 61672262), Shandong Provincial Key Research and Development Project (nos. 2019GGX101028 and 2018CXGC0706), Shandong Provincial Natural Science Foundation (nos. ZR2021LZH007 and ZR2019LZH015), Shandong Province Higher Educational Science and Technology Program (no. J18KA349), and Project of Independent Cultivated Innovation Team of Jinan City (no. 2018GXRC002).

References

- [1] Y. Zhao, R. Yang, G. Chevalier, and M. Gong, "Deep residual bidir-lstm for human activity recognition using wearable sensors," *CoRR, abs*, vol. 1708, Article ID 08989, 2017.
- [2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and y. A. Blaise Aguera, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference On Artificial Intelligence And Statistics*, A. Singh and J. Zhu, Eds., vol. 54, pp. 1273–1282, PMLR, Lauderdale, FL, USA, April 2017.
- [3] M. J. Sheller, G. Anthony Reina, B. Edwards, J. Martin, and S. Bakas, "Multi-institutional deep learning modeling without sharing patient data: a feasibility study on brain tumor segmentation," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, A. Crimi, S. Bakas, H. Kuijff, F. Keyvan, M. Reyes, and T. van Walsum, Eds., Springer International Publishing, Berlin, Germany, pp. 92–104, 2019.
- [4] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," 2017, <https://arxiv.org/abs/1710.06963>.
- [5] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "Psspr: a source location privacy protection scheme based on sector phantom routing in wsns," *International Journal of Intelligent Systems*, vol. 37, 2021.
- [6] B. Keith, V. Ivanov, B. Kreuter et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, Dallas, TX, USA, October 2017.
- [7] R. Kumar, A. A. Khan, J. Kumar et al., "Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging," *IEEE Sensors Journal*, vol. 21, no. 14, pp. 16301–16314, 2021.
- [8] Y. Qi, M. S. Hossain, J. Nie, and X. Li, "Privacy-preserving blockchain-based federated learning for traffic flow prediction," *Future Generation Computer Systems*, vol. 117, pp. 328–337, 2021.
- [9] H. Khelifi, S. Luo, B. Nour et al., "Bringing deep learning at the edge of information-centric internet of things," *IEEE Communications Letters*, vol. 23, no. 1, pp. 52–55, 2018.
- [10] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge

- computing in real-time industrial informatics,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, 2019.
- [11] C. Zhao, S. Zhao, M. Zhao et al., “Secure multi-party computation: theory, practice and applications,” *Information Sciences*, vol. 476, pp. 357–372, 2019.
- [12] “How to share a secret,” *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [13] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, “Semi-selfish mining based on hidden Markov decision process,” *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
- [14] G. Wood, P. Ardoin, D. M. Brink et al., “Ethereum: a secure decentralised generalised transaction ledger,” *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.
- [15] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, “Globally distributed content delivery,” *IEEE Internet Computing*, vol. 6, no. 5, pp. 50–58, 2002.
- [16] M. Armbrust, A. Fox, R. Griffith et al., “A view of cloud computing,” *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [17] M. Mohammed Sadeeq, N. M. Abdulkareem, S. R. Zeebaree et al., “Iot and cloud computing issues, challenges and opportunities: a review,” *Qubahan Academic Journal*, vol. 1, no. 2, pp. 1–7, 2021.
- [18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and y. A. Blaise Aguera, “Communication-efficient learning of deep networks from decentralized data,” pp. 1273–1282, PMLR, 2017, <https://arxiv.org/abs/1602.05629>.
- [19] Li Tian, A. Kumar Sahu, A. Talwalkar, and V. Smith, “Federated learning: challenges, methods, and future directions,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.
- [20] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, “Is semi-selfish mining available without being detected?” *International Journal of Intelligent Systems*, 2021.
- [21] N. Szabo, “Formalizing and securing relationships on public networks,” *First Monday*, vol. 2, 1997.
- [22] W. Zou, D. Lo, P. Singh Kochhar et al., “Smart contract development: challenges and opportunities,” *IEEE Transactions on Software Engineering*, vol. 47, 2019.
- [23] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of COMPSTAT’2010*, pp. 177–186, Springer, Paris, France, August 2010.
- [24] A. Kate, G. M. Zaverucha, and I. Goldberg, “Constant-size commitments to polynomials and their applications,” in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 177–194, Springer, Singapore, December 6–10, 2021.

Research Article

Multiple-Layer Security Threats on the Ethereum Blockchain and Their Countermeasures

Li Duan ^{1,2}, Yangyang Sun ¹, Kejia Zhang ^{3,4} and Yong Ding ²

¹Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China

²Guangxi Key Laboratory of Cryptography and Information Security, Guilin, Guangxi, China

³School of Mathematical Science, Heilongjiang University, Harbin 150080, China

⁴Cryptology and Cyberspace Security Laboratory of Heilongjiang University, Harbin 150080, China

Correspondence should be addressed to Li Duan; duanli@bjtu.edu.cn and Kejia Zhang; zhangkejia.bupt@gmail.com

Received 19 November 2021; Revised 10 January 2022; Accepted 20 January 2022; Published 14 February 2022

Academic Editor: Yuling Chen

Copyright © 2022 Li Duan 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 technology has been widely used in digital currency, Internet of Things, and other important fields because of its decentralization, nontampering, and anonymity. The vigorous development of blockchain cannot be separated from the security guarantee. However, there are various security threats within the blockchain that have shown in the past to cause huge financial losses. This paper aims at studying the multi-level security threats existing in the Ethereum blockchain, and exploring the security protection schemes under multiple attack scenarios. There are ten attack scenarios studied in this paper, which are replay attack, short url attack, false top-up attack, transaction order dependence attack, integer overflow attack, re-entrancy attack, honeypot attack, airdrop hunting attack, writing of arbitrary storage address attack, and gas exhaustion denial of service attack. This paper also proposes protection schemes. Finally, these schemes are evaluated by experiments. Experimental results show that our approach is efficient and does not bring too much extra cost and that the time cost has doubled at most.

1. Introduction

In recent years, with the rapid development of blockchain technology, the application scenarios of blockchain have not only been limited to digital currency and financial fields but have gradually been deeply integrated with all walks of life [1, 2], such as smart city and Internet of things (IoT). In 2008, Satoshi Nakamoto released his famous Bitcoin whitepaper [3], which first put forward the concept of “blockchain.” Blockchain is a new distributed computing and storage paradigm which integrates many existing technologies. It uses cryptography principle and timestamp technology in data layer to ensure the immutability of data, uses peer-to-peer network to communicate data in network layer, uses distributed consensus algorithm to maintain the consistency of data in the consensus layer; uses scripts and algorithms to implement smart contracts in contract layer; and uses Turing complete virtual machine to realize various functions in the application layer. Compared with

traditional databases, blockchain, as a distributed database, requires multiple nodes to maintain data together, which requires data consistency and business fairness.

At the end of 2013, Vitalik Buterin, founder of Ethereum, released the first edition of Ethereum White Paper [4], which realized the development of smart contracts with Turing’s complete programming language. From then on, blockchain application was no longer limited to the currency field, and the blockchain 2.0 era started. As an open source public chain platform, Ethereum’s function of supporting smart contracts will help its development. Smart contract is a representative technology in the blockchain 2.0 era, and its concept was put forward by cryptographer Szabo [5] as early as the end of the 20th century. He defined smart contract as a set of promises defined in digital form, and the participants of the contract can implement these promises on machines. It was limited to the science, technology, and environment at that time, and it was not until the birth of Ethereum that it gradually revived.

Blockchain technology, as a new technology, technically ensures transaction security through encryption algorithm and digital signature, and relies on consensus mechanism to generate blocks to form a chain structure to ensure that data cannot be tampered with. Nevertheless, blockchain is still facing great security threats [6], especially in smart contracts. Because of the differences in the programming ability of smart contract developers, security problems are inevitable. On June 18, 2016, hackers maliciously attacked The DAO project, resulting in the theft of 3.6 million Ether and the loss of nearly 100 million funds. On July 20, 2017, hackers exploited the contract loophole of Parity Multi-Signature Library, resulting in the freezing of over 500,000 Ether in 587 wallets and a loss of about RMB 220 million. In April 2018, nearly RMB 6 billion was stolen by hackers due to integer overflow loopholes in the contract code of American Chain BEC project, which reduced the market value of tokens to almost zero. In 2019, the global blockchain lost more than \$6 billion due to security incidents. In 2020, the blockchain was hacked incurring a loss of nearly \$3.8 billion.

Therefore, it is meaningful to study the security attacks of blockchain, especially to study the security threats of smart contracts that cause the greatest losses, which is helpful to improve the security level of the Ethereum blockchain. The contributions of this paper are summarized as follows:

- (i) We introduce the background of various attacks, and analyze the principles and attack paths of ten kinds of security threats on Ethereum.
- (ii) We construct several specific attack scenarios, and propose the protection schemes corresponding to Ethereum attacks.
- (iii) We test and evaluate the proposed protection schemes. Finally, a demonstration system is built to demonstrate the multiple attack scenario.

The rest of this paper is organized as follows. Section 2 reviews the related work. The backgrounds and architecture of the Ethereum blockchain are introduced in Section 3. In Section 4, we study the principle of ten kinds of attacks on Ethereum. In Section 5, we explore the corresponding protection schemes. The protection schemes are evaluated in Section 6. In Section 7, this paper is concluded.

2. Related Work

Aiming at the security threats in blockchain scenarios, the existing research work mainly focuses on attack discovery and attack protection.

2.1. Attack Discovery. From the perspective of attack discovery, the detection methods based on symbol execution, fuzz testing, taint analysis, and formal verification are used to monitor the security threats of contract generation, release, and execution, and to detect the potential risks and vulnerable paths in the process of contract interaction. Luu et al. [7] proposed a detection method based on symbolic model to monitor the security threats in the whole process of contract generation-release-execution in real time, and to

detect the potential risks in the process of contract interaction and the vulnerability of accurate location of vulnerable paths. Its design is fully modularized, allowing advanced level users to execute and plug in their own identification logic to check self-defined properties in their smart contracts. In addition, there are many automated detection tools, for example, teEther [8], Securify [9], ZEUS [10], EasyFlow [11], and SmarTest [12]. There are many detection tools at present, but they are difficult to be widely used. Tu et al. [13] proved that the detection efficiency is not high, and there are fewer vulnerabilities that can be detected. We can combine traditional detection methods with machine learning to improve the versatility and efficiency of detection tools to a certain extent.

In addition, Hou et al. [14] put forward the method of deep reinforcement learning by analyzing the behavior of associated users, and automatically discovering the attack of consensus strategy. Li et al. [15] proposed an improved selfish mining based on hidden Markov decision processes to maintain the benefit from selfish mining. Li et al. [16] focused on the validity of semi-selfish mining attacks considering the probability of being detected. Marcus et al. [17] put forward a method of solar eclipse attack on Ethernet network with very few resources.

2.2. Attack Protection. From the perspective of attack protection, based on multi-signature, Byzantine consensus virtual layer design, and safe miner selection, the problems of DoS attack and currency age attack in blockchain operation are solved. Li et al. [18] proposed a cross-chain system based on multiple signatures, which can ensure the credibility of trading groups by locking assets and resisting DoS attacks at the same time. Sonnino et al. [19] proposed a cross-ledger consensus protocol based on Byzantine consensus mechanism to resist cross-ledger replay attacks. Li et al. [20] used the amount of coins to select miners and limit the maximum value of currency age to fight against currency age attacks, thus improving the robustness of the system. Wang et al. [21] proposed a secure inter-chain transmission protocol, which can effectively resist the double-flower attack by recording the asset transmission process between multiple chains and ensuring its consistency. Luu et al. [22] put forward the verifier dilemma problem, that is, after the nodes participating in the consensus pay a lot of computing power to verify the transaction, if they do not get the bookkeeping right, the verifier will face the dilemma of paying more computing power to verify or accepting the wrong script, so as to solve the double-flower attack problem. Luu et al. [23] put forward a secure fragmentation protocol which can be used to build public chains, and the analysis proves that this scheme can effectively improve the system throughput. Nguyen et al. [24] proposed an approach and a tool, called SGUARD, which automatically patches vulnerable smart contracts.

As a complex system, blockchain faces security threats from the data layer to the application layer. At present, the related work of blockchain security attack and protection is mainly discussed from the attack as a whole, but not the

specific attack. However, this paper goes deep into the details, investigates the security problems faced by Ethereum at all levels, studies and tests several typical attacks existing in smart contracts, and proposes protection schemes.

3. Background

The related technologies and background knowledge involved in this section are introduced, including Ethereum architecture, memory layout, and transaction process.

3.1. Ethereum. Ethereum is an open-source public chain platform for executing intelligent contracts through Ethereum virtual machines. These machines execute intelligent contracts by consuming Ethereum coins. The concept of Ethereum first appeared at the end of 2013. Inspired by Bitcoin, Vitalik Buterin, founder of Ethereum, released the first edition of Ethereum white paper, which realized the development of intelligent contracts with Turing’s complete programming language.

As of November 2021, the market value of Ethereum has exceeded \$570 billion, which is the second highest cryptocurrency in market value after Bitcoin with \$1.27 trillion. Ethereum is often described as “the computer of the world.” From the point of view of computer science, Ethereum is a deterministic but unbounded state machine, which has two basic functions, the first is a globally accessible singleton state, and the second is a virtual machine that changes the state. It uses blockchain to synchronize and store the state of the system [25], and cryptocurrency called Ether is used to calculate and limit the execution resource cost. Ethereum developers can write intelligent contracts and build decentralized applications that can run on Ethereum virtual machines. While ensuring stable and normal operation, it can also reduce or eliminate examination procedures, and save resources and reduce risks by eliminating the participation of third parties.

3.2. Ethereum Architecture. As Figure 1 shows, Ethereum architecture is composed of five layers [26], namely, the data, network, consensus, contract, and application layers. Ethereum system runs on these five layers. The data layer includes technical elements such as data block, chain structure, hash function, asymmetric encryption, timestamp, Merkle tree, etc., which ensures the reliability and stability of Ethereum data [27]. The network layer specifies the peer-to-peer network, wherein each node can obtain the updated status of blockchain from some active nodes. There is no central server, and only the nodes exchange information fairly. The consensus layer ensures the consistent state of the blockchain. At present, Ethereum adopts the Proof of Work (PoW) consensus mechanism. But in the future version planning, Ethereum consensus mechanism will gradually transition to Proof of Stake (PoS) mechanism [28]. This design can speed up the transaction and save the resource consumption [29]. It is also effective to avoid the disadvantage of unfair initial equity distribution existing in the simple equity proof mechanism. The contract layer

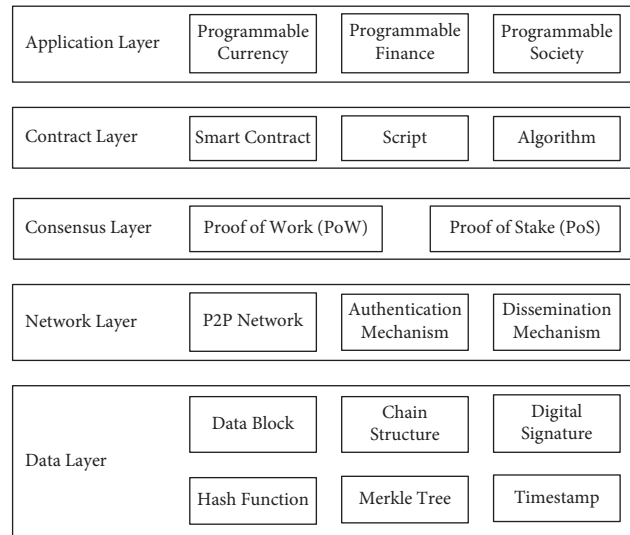


FIGURE 1: Ethereum’s hierarchical structure.

encapsulates various scripts, algorithms, and smart contracts, so that various instructions can be executed automatically and determinately. Smart contract is executed in Ethereum virtual machine at a certain cost of gas according to different instructions. A smart contract is also an Ethereum account, which we call a contract account (CA). This means that they have a balance and they can trade through the network. But they cannot be manipulated by humans. They are deployed on decentralized network nodes and run as programs. Individual users can interact with the smart contract by submitting a transaction to execute a certain function. Smart contracts can define rules like regular contracts and automatically enforce them through code. The application layer encapsulates various application scenarios and cases. For example, various blockchain applications built on Ethereum are deployed in the application layer. And, it is the basis for the realization of a programmable society in the future. Finally, some corresponding components are needed to serve these five layers, which is called external necessary environment, such as web user interface interacting with applications, database for storing blockchain data, cryptographic mechanism supporting consensus protocol, etc. [30].

3.3. Memory Layout of Ethereum. Ethereum virtual machine (EVM) is Turing-complete, and its operations are limited by the number of gas provided by users for each transaction. The implementation of the Ethereum virtual machine is based on the stack. Unlike traditional computers, all instructions of Ethereum virtual machine are executed on the stack, and the parameters or operation results required by the instructions can be obtained through the stack operation. The maximum depth of the Ethereum virtual machine stack is 1024, and the size of each data unit in the stack is 256 bits, which is convenient for executing Keccak-256 elliptic curve hashing algorithm. There are two main storage models in the Ethereum virtual machine, namely, temporary memory

model and permanent storage model. Temporary memory of virtual machine is a simple byte array based on word addressing, which is similar to the concept of traditional computer memory, and its storage is unstable. Unlike temporary memory, permanent storage is a word array based on word addressing. As a part of the system state, permanent storage will be maintained in real time, and it is more stable than its temporary memory. In the initial state, both the data in the temporary memory area and the data in the permanent storage area of Ethereum virtual machine are initialized to 0. Storage stores data through key-value pairs, which maps 32-byte keys to 32-byte data. Global variables in smart contracts are stored in the storage area, and their storage location is determined by the type of the variable and its position in the code. If a variable store is less than 256 bits, Ethereum virtual machine may store multiple variables in one slot. Or the mapping type occupies a slot in which the mapping or array length is stored. The specific locations of arrays and mapping elements are stored in slots according to a set of special hash rules.

3.4. Transaction Process of Ethereum. The trading process of Ethereum is as follows. (1) The sender constructs a transaction and digitally signs it. (2) The sender calls api through JSON-RPC to submit the signed transaction to the Ethereum client. (3) After verifying the received transaction, the Ethereum client broadcasts it to Ethereum point-to-point network. (4) Any client that receives the transaction information will add the transaction to its transaction pool if the client is also a miner. (5) The miner executes a series of transactions selected from its trading pool, creates a new block, and updates the status of the block chain. There are three types of transactions. For transfer transactions, the specified amount needs to be updated and transferred from the sender's account to the receiver's account or contract account. For contract deployment, enter a bytecode to create a new contract account and associate it with the entered bytecode. For contract call, where the recipient is the called smart contract, the input uniquely identifies the callee function through the hash digest algorithm, and the bytecode associated with the called smart contract account is loaded into the Ethereum virtual machine for execution. (6) Miners solve the problem of workload proof by looking for a random nonce value. The hash value of metadata of this new block needs to be smaller than a certain value, which reflects the difficulty of creating the block. Unlike Bitcoin's computationally intensive workload, Ethereum uses a memory-intensive problem called "Ehash." (7) When creating the block, the miners broadcast it to the point-to-point network of Ethereum, so that other clients can verify the block. (8) When other Ethereum clients verify a new block, the client will add the block to the blockchain.

4. Security Attacks on Ethereum

Attacks on Ethereum can be divided into five layers: application layer, contract layer, consensus layer, network layer, and data layer attacks. In this paper, we focus on the

attacks in the application layer, the contract layer, and the network layer. The other two layers of attacks are our future research directions.

4.1. Attacks of Application Layer. The application layer is the carrier of blockchain technology and provides solutions for various business scenarios. Security vulnerabilities in various trading platforms and user accounts seriously threaten the asset security of blockchain wallet users. Therefore, we analyze three common types of attacks.

4.2. Replay Attack. The replay attack is to replay transaction information. The user signs a message, uploads it to the contract, and then verifies the signature inside the contract. But since the user's signature information is online, everyone can get it. When verifying the user's signature in the contract, if the signed message does not include variables that change randomly with the number of transactions, such as timestamp, nonce, etc., the attacker will hold the user's signature and forge transactions, thereby obtaining a profit. It can be widely understood as the process of using the same payment information to purchase goods multiple times. When the Ethereum and Ethereum Classic chains emerged after the hard fork, it was found that transactions on the Ethereum chain were still valid when they were replayed on the Ethereum Classic chain. As Figure 2 shows, while parameters remain unchanged, multiple transfers can be made through the replay attack.

4.3. False Top-Up Attack. The status field in the Ethereum token transaction receipt is true or false depending on whether an exception is thrown during the execution of the transaction. When the user calls the transfer function of the token contract to transfer, if the transfer function runs normally and no exception is thrown, the status of the transaction is true. If digital currency exchanges, wallets, and other platforms have flaws in determining whether tokens' recharge transactions are successful, it will lead to serious false top-up attack. As Figure 3 shows, when $\text{balances[msg.sender]} < _value$, it enters the else logical section and returns false, and finally no exception is thrown. In this attack, although the exchange did not receive the real tokens, the transaction execution did not throw an exception, and the user did get the real recharge record. In this case, users can steal real assets. The false top-up attack has become a type of attack that cannot be ignored in blockchain system.

4.4. Transaction Order Dependence Attack. Transaction order dependence attack is a kind of attack that widely exists in the blockchain system; an example of transaction order dependence attack is shown in Figure 4. In blockchain, transactions initiated by nodes need to be packaged by miners before they can be finally recorded on the blockchain. Miners select a series of transactions from the trading pool and then package them into a new block. According to

```
function transferProxy(address _from, address _to, uint256 _value, uint256 _fee,
    uint8 _v, bytes32 _r, bytes32 _s) public returns (bool){
    bytes32 h = keccak256(_from,_to,_value,_fee);
    if(_from != ecrecover(h,_v,_r,_s)) revert();
}
```

FIGURE 2: Replay attack.

```
function transfer1(address _to, uint256 _value) returns (bool) {
    if(_value <= balance[msg.sender] && _value > 0)
    {
        balance[msg.sender] -= _value;
        balance[_to] += _value;
        return true;
    }
    else
        return false;
}
```

FIGURE 3: False top-up attack.

```
event Purchase(address _buyer, uint256 _price);
event PriceChange(address _owner, uint256 _price);
modifier ownerOnly() {
    require(msg.sender == owner);
    _;
}
function TransactionOrdering() {
    owner = msg.sender;
    price = 100;
}
function buy() returns (uint256) {
    Purchase(msg.sender, price);
    return price;
}
function setPrice(uint256 _price) ownerOnly() {
    price = _price;
    PriceChange(owner, price);
}
```

FIGURE 4: Transaction order dependence attack.

miners' criteria for transaction selection, miners will generally choose the transaction fees to sort and package in order to get the maximum benefits. Therefore, the sequence of a series of transactions packaged in the block is not the same as the sequence of transaction generation but is also related to the gas cost consumed by the transaction. Therefore, the contract code cannot know the order of transactions. And, the transaction is visible to each node in the transaction pool, so its execution order can be observed.

The attacker observes the transactions that may contain the target contract in the pool. If they exist, the status of the contract that is not conducive to the attacker or the authority of the contract will be modified by the attacker. Attackers can also steal transaction data, create their own transactions

at a higher gas price, and then package their own transactions in the block before the original transaction, thus obtaining transaction processing priority. In Ethereum geth client, txpool consists of two parts, namely, pending queue and queued queue. When the sending transaction Nonce is greater than the completion transaction nonce+1, the transaction will be queued, and if the current sending transaction nonce is equal to the completion transaction nonce + 1, the transaction will be placed in pending waiting for packaging.

4.5. Attacks of Contract Layer. As an indispensable part of blockchain technology, smart contract not only expands the application of blockchain technology but also increases the attack surface faced by the blockchain system. The smart contract is written in a high-level language like solidity, and then the contract will be compiled into bytecode, which will be deployed to the blockchain by the contract owner and run on various virtual machines similar to Ethereum virtual machines. In the process, the smart contract will face various security threats [31].

4.6. Integer Overflow Attack. Integer overflow is a typical loophole in the blockchain system, which once caused serious economic losses in the development of blockchain. In the Ethereum platform, Solidity language is the most mainstream language for writing intelligent contracts. Because of the insecurity of its design, integer overflow is a serious problem. Generally speaking, integer overflow can be divided into integer overflow and integer underflow. According to arithmetic classification, there are three overflow problems: multiplication overflow, addition overflow, and subtraction overflow. In April 2018, nearly RMB 6 billion was stolen by hackers due to integer overflow loopholes in the contract code of the American Chain BEC project, which reduced the market value of tokens to almost zero. In the same month, hackers used the integer overflow vulnerability of SMT project side to create a huge amount of SMT currency for selling, and the Firecoin Exchange suspended the recharge and withdrawal of all other currencies for this purpose.

In Solidity, the variable supports unsigned integers, and the value after uint represents the number of bits occupied by its unsigned integers in storage, and supports 8-bit unsigned integers to 256-bit unsigned integers. An unsigned integer of type uint8 stored in the range of 0 to $2^8 - 1$, that is, [0, 255], and an unsigned integer of type uint256 stored in the range of 0 to $2^{256} - 1$. Because the range of stored integers from uint8 to uint256 is limited, and the range of represented integers is also limited, there is an overflow problem. The integer overflow attack is shown in Figure 5. When $balances[msg.sender] < _amount$, it results in an underflow.

4.7. Re-Entrancy Attack. Re-entrancy attack is a typical attack in Ethereum, which directly led to the hard bifurcation of Ethereum. The main reason for the attack is the sequencing and atomicity of updating smart contract

```
function withdraw(uint _amount) {
    balancesAndAmount(balances[msg.sender], _amount);
    require(balances[msg.sender] - _amount > 0);
    msg.sender.transfer(_amount);
    balances[msg.sender] -= _amount;
}
```

FIGURE 5: Integer Overflow attack.

variables and transferring operations, the re-entrancy attack is shown in Figure 6. When the logic in the smart contract code adopts the sequence of transferring operation first and then modifying the variable value, the attacker can construct a smart contract with the malicious callback function. If the object of the transfer operation is a malicious contract, it can lead to recursively calling the contract, destroying the original business logic of the contract, and bypassing its inspection to obtain additional transfer income.

By default, the Ethereum smart contract has an unnamed callback function, which has no parameters or return values. If no function can be found in the calling contract to match the hash of the provided function, the callback function will be called. When the contract receives a transfer without data, it will also call the callback function. In addition, in order to receive Ether, the callback function must be marked as payable. If it is not marked as payable, the contract can only receive Ether by calling other functions marked with payable. Imagine such a scenario, if a special callback function is constructed, in which the transfer function of the other party is called, then a recursive transfer will be generated, and the contract with loopholes will continuously transfer money to the special contract until the gas is exhausted. It should be noted that this attack is only aimed at the transfer method of `address.call.value ()` in Ethereum solidity.

4.8. Honeypot Attack. Honeypot contracts are the most interesting findings. These contracts hold ether, and pretend to do so insecurely. In short, they are scam contracts that try to fool us into thinking we can steal the ether they hold, while in fact all we can do is lose ether. As Figure 7 shows, CryptoRoulette is a type of honeypot attack. The variable of game is not initialized, so it by default points to the first location of the contract storage space, and then stores the caller's address here. The submitted number is stored in the second location. In fact, the variable of `secretNumber` eventually is overwritten by the address of the caller's. A common pattern they follow is, in order to win the ether they hold, we must send them some ether of our own first. However, if we try that, we are in for a nasty surprise: the smart contract eats up our ether, and we find out that the smart contract does not do what we thought it would.

4.9. Short Url Attack. Short url attack is a typical attack in Ethereum, which usually occurs in exchanges. In Ethereum virtual machine, the data end of the input will be automatically filled with 0. Malicious attackers can use an address account with the end of 0, and the exchange fails to verify the address length

```
contract Victim{
    function withDraw(){
        uint amount = userBalannce[msg.sender];
        if (amount > 0) {
            msg.sender.call.value(amount());
            userBalannce[msg.sender] =0;
        }
    }
}
contract Attacker{
    function() payable{
        test++;
        Victim(msg.sender).call(bytes4(keccak256("withDraw()")));
    }
}
```

FIGURE 6: Re-entrancy attack.

```
struct Game {
    address player;
    uint256 number;
}
Game[] public gamesPlayed;
function shuffle() {
    secretNumber = uint8(sha3(now, block.blockhash(block.number-1)))% 10;
}
function play(uint256 number) payable public {
    require(msg.value >= betPrice && number <= 10);
    Game game;
    game.player = msg.sender;
    game.number = number;
    gamesPlayed.push(game);
    if (number == secretNumber) {
        msg.sender.transfer(this.balance);
    }
    shuffle();
    lastPlayed = now;
}
```

FIGURE 7: Honeypot attack.

input by the user, which causes the transferred related variables to shift and enlarge, thus expanding the actual transfer amount by several times, and malicious attackers can obtain a large amount of benefits. There are two main reasons for this vulnerability; one is that the exchange has not verified the incoming address length of the user, and the other is that the Ethereum virtual machine has an automatic completion mechanism for the data whose length does not conform to the specification when calling the smart contract, resulting in the shift amplification of parameters. We can use `sendRawTransaction()` to achieve this attack and the code is shown in Figure 8.

4.10. Airdrop Hunting Attack. The airdrop hunting attack uses multiple new accounts to call the airdrop function in order to obtain airdrop coins, and attackers transfer them to

```

mapping (address => uint) balances;
event Transfer(address indexed _from, address indexed _to, uint256 _value);
function transfer(address to, uint amount) public returns(bool success) {
    if (balances[msg.sender] < amount) return false;
    balances[msg.sender] -= amount;
    balances[to] += amount;
    emit Transfer(msg.sender, to, amount);
    return true;
}

```

FIGURE 8: Short URL attack.

their account to achieve wealth accumulation. This attack is relatively common that as long as it is a contract with an airdrop function, it can make multiple profits. The first automated attack was the Simoleon contract. As Figure 9 shows, the contract was designed to give some amount of ether to initialized an account, so the attacker thinks that we can create a few more accounts to get rewards, then transfer all the money to one account. The attacker write attack the contract and create many temporary contracts, and call this function in these contracts.

4.11. Writing of Arbitrary Storage Address Attack. The attack of arbitrary memory address writing is a common and harmful attack in the blockchain system. The attack can cause malicious users to write and overwrite any storage variable in the smart contract. In Ethereum, the state variables of intelligent contracts will be stored in the storage area, which is an important and open contract storage space. Generally speaking, contract developers will set strict access control to the global variables stored in the storage area to ensure the security of contracts. Storage key-value pair mapping is used to store data. If the user can arbitrarily control the key value of storage when writing, he or she can modify any storage variable value, so as to avoid all the related detection operations in the contract that uses the state variable value to check the authority, and thus achieve the purpose of improving the authority. In addition, because the attacker can use this vulnerability to destroy the contract storage structure, and perform any variable overwriting operation, such as overwriting the value of the state variable storing the address of the contract owner, this may cause abnormal execution of contract functions, freezing of funds, and other hazards. Since the required guard is invalid, the contract owner can try to underflow the array size by executing the code of Figure 10 when the array length `bonusCodes` is 0. Therefore, we can write to any location in the storage arbitrarily.

4.12. Attacks of Network Layer. The network layer is the most basic technical architecture in the blockchain system. It encapsulates the blockchain system's networking methods, message dissemination mechanisms, authentication mechanisms, etc., so that the blockchain has decentralized and nontamperable characteristics. But these features also

```

function transfer(address _to, uint256 _amount) returns (bool success) {
    initialize(msg.sender);
    if (balances[msg.sender] >= _amount
        && _amount > 0) {
        initialize(_to);
    } else {
        return false;
    }
}

function initialize(address _address) internal returns (bool success) {
    if (_totalSupply < _cutoff && !initialized[_address]) {
        initialized[_address] = true;
        balances[_address] = _airdropAmount;
        _totalSupply += _airdropAmount;
    }
    return true;
}

```

FIGURE 9: Airdrop hunting attack.

```

function PopBonusCode() public {
    require(0 <= bonusCodes.length);
    bonusCodes.length--;
}

function UpdateBonusCodeAt(uint idx, uint c) public {
    require(idx < bonusCodes.length);
    bonusCodes[idx] = c;
}

```

FIGURE 10: Writing of arbitrary storage address attack.

provide convenience for attackers who can easily launch a DoS attack. The purpose of the attack is to make users temporarily or permanently unable to use these services provided by the smart contract.

4.13. Gas Exhaustion Denial of Service Attack. According to the design of Ethereum, when the smart contract is deployed or the function in the smart contract is called, the execution of the contract code needs a certain amount of gas to ensure that the calculation is completed completely. At the same time, the Ethereum system limits the maximum total amount of gas consumed by each block, and the total amount of gas of all transactions in the block cannot exceed the maximum total amount of gas in this block. Once an operation in an intelligent contract consumes a lot of gas, resulting in the consumed gas value reaching the maximum total amount of gas in the block, the operation will not be successfully executed, and all processes depending on the operation will fail, so the contract cannot normally complete other functions, resulting in a denial of service state. As Figure 11 shows, transferring money to everyone at once is likely to result in reaching the gas limit of ethereum blocks. Usually, this denial of service attack occurs when a contract

```

address public owner;
address[] investors;
uint[] investorTokens;
function invest() public payable {
    investors.push(msg.sender);
    investorTokens.push(msg.value * 5);
}
function distribute() public {
    require(msg.sender == owner);
    for(uint i = 0, i < investors.length; i++) {
        transferToken(investors[i], investorTokens[i]);
    }
}

```

FIGURE 11: Gas exhaustion denial of service attack.

developer does not consider the block gasLimit and introduces the operation of modifying dynamic data structure variables such as arrays whose size will change with time. After a block is mined, an attacker can issue multiple transactions at a higher gas price immediately, and then use the above operations of the contract to consume the gas limit of the whole block, so that the block does not contain any other transaction before a certain time, thus preventing other users from using the functions of the contract normally.

5. Security Protection Schemes

In this section, we propose the protection schemes against the ten attacks mentioned in the previous section. The details follow.

5.1. Protection Schemes of the Application Layer. We can prevent the replay attack in the following ways: (1) Avoiding using the transferProxy function and using a more secure signature method. (2) Adding variables such as nonce, timestamp, etc. The nonce generation algorithm does not adopt the design of self-increment from 0 to avoid the same value as other scenarios. (3) Adding address (this) in keccak256(). (4) Adding chainID, which is the blockchain's name.

To prevent the false top-up attack, we judge not only transaction success but also whether the balance of the top-up wallet address increases accurately. This judgment can be made through the Event log. Many centralized exchanges, wallets, and other service platforms obtain the transfer amount and judge the accuracy of the transfer through Event logs. However, we need to pay special attention to the evil situation of the smart contract, because the Event can be written arbitrarily, and it is not a mandatory default option that cannot be tampered with. The required and asserted methods can also be used that an exception will be thrown directly to interrupt the execution of the subsequent instructions of the contract when the conditions are not met.

The protection of transaction order dependence attack is a very complicated process. For the ERC20 transaction order

dependence attack that happened once, it only needs the contract developers to pay attention to this problem and follow the best programming practices. For the attack scenario constructed in this example, this problem is not the problem of the contract developer, but the problem of the Ethereum system itself. At present, the better solution is to confuse transactions, such as hiding transactions as internal transactions, and so on.

5.2. Protection Schemes of Contract Layer. For the problem of integer overflow, we can consider the results of each step by setting up a complete inspection mechanism, but this method is difficult and cumbersome, and it is not universal. Therefore, OpenZeppelin provides SafeMath [32] in an intelligent contract function library, which can effectively prevent integer overflow. There are two ways to use the SafeMath library. The first one is to use the library functions directly, such as SafeMath.add(a,b). The other is that library functions can be called after using SafeMath for unit. For example, a.add(b) means that add(a,b) in safemath library has been executed.

For the protection of re-entrancy attack, the most fundamental solution is to update all the states that should be changed in advance before the transfer, instead of updating them after the transfer, which depends on the smart contract developers to follow the best practices. In addition, it is also an idea to use other transfer methods instead of the msg.sender.call.value() function. For the designed attack scenarios, we use these two methods to test them, respectively. For the first method, we put the change in account balance before the transfer, and then judge whether the transfer is successful or not, and if the transfer is not successful, restore the balance of the user's account. In this way, the code re-entrancy attack is successfully prevented, and the protection scheme is effective. For the second scheme, we use the transfer() function to replace the msg.sender.call.value() function, which can also prevent the re-entrancy attack. The above two schemes can well prevent a re-entrancy attack, but the best scheme is the first one, which updates the status first and then transfers money.

Honeypot contracts are diverse and unpredictable. For the CryptoRoulette attack, we can clearly use memory or storage for variables. We can also use the new version of the compiler with version 0.5.0 and later where this problem has been solved by the system because smart contracts with uninitialized storage variables cannot be successfully compiled. Finally, we remind everyone that some people use Ethereum smart contracts to cheat. They fully figure out the psychology of some people's greed for small profits, and throw out some seemingly handy bait, then run away after having enough users. Because these creators spend for fees to create these contracts, they have a purpose that putting a certain amount of ether can get all the balance of the account, so it is definitely arbitrage. Publishing the source code on Github also uses various tricks to make people not find loopholes in a short time, thus encouraging users to enter the trap.

Short url attack protection only needs the exchange to increase the address length check at the client. In addition, for contract developers, the web3 interface used has already fixed the vulnerability. When users call the contract with web3, if they find that the data length is insufficient, they will not add 0 at the end, but add 0 at the beginning of the field, which effectively prevents the short url attack. In a word, the protection of this vulnerability mainly depends on two parts, one is that the client actively checks the address length, the other is that the parameter format check is added at the web3 level. Although this vulnerability can be reproduced at the virtual machine level of Ethereum, there will be no problem in the actual application scenario of the blockchain.

To prevent an airdrop hunting attack, we can set permission control for the airdrop function. For example, only the contract creator can distribute tokens to target addresses. Or only externally owned accounts can receive airdrop rewards, and contract accounts cannot participate.

For any memory address write attack, this attack is rare, and it is often the result of many factors. Therefore, the protection of this attack can be achieved by the contract developers following the best practices. In the development of contracts, developers need to pay attention to dynamic arrays. Errors in the processing of dynamic arrays may lead to contract loop-holes in an unobvious way. Therefore, in unnecessary cases, dynamic array is not used, which can effectively avoid this attack.

5.3. Protection Schemes of the Network Layer. Gas exhaustion denial of service attack protection also depends on the best practices of contract developers. The size of the gas consumed by different instructions is not certain. By debugging the attack scenario, it is found that the load instruction was executed in the loop, consuming 800 gas. However, the operation with high gas consumption is usually to operate the data in the storage area, so the contract developer should try not to operate the data in the storage area in the loop. Besides, we can also add an end mark of the loop in the execution.

6. Program Evaluation

6.1. Experiment Setup. The private Ethereum blockchain is deployed on Alibaba cloud server, which has 4 processors with 8 GB RAM and 200 GB hard Disk, and each processor has 2 cores. The server is running with Ubuntu 18.04. Smart contracts are written by Solidity programming language.

6.2. Experiment Processes. Based on the above configuration, we implemented ten defense methods as mentioned in the previous section. To analyze their efficiency, we tested time cost of 50, 100, 150, 200, 250, and 300 transactions. The experimental results are shown in Figure 12.

6.3. Result Analysis. In Figure 12, we find the most time-consuming is the replay attack's protection scheme, followed

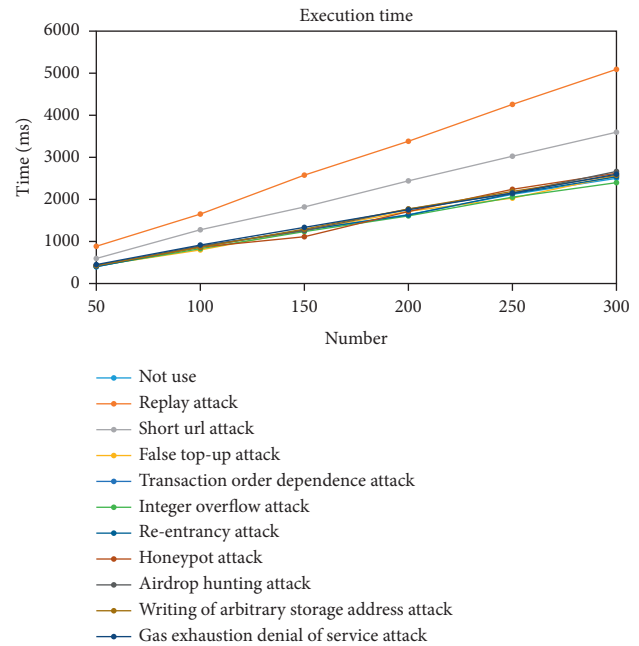


FIGURE 12: Time cost with different protection schemes.

by the short url attack's protection scheme. The replay attack's protection scheme adds signing and signature verification, which need to perform complex calculations, so it is the most time-consuming. For the short url attack's protection scheme in web3, if the data length is insufficient, it will add 0 at the beginning of the field. First judge the address length, if the length is less than 40 bit, then web3 calls a function to automatically complement the address which is time-consuming. Even with these two schemes, the time cost has doubled at most. The time cost of the other eight protection schemes is roughly the same as transactions without them, and can be ignored. Because they either change the execution order of the code or add a judgment, they do not bring too much extra cost. All in all, these protection schemes do not bring much time cost and they are efficient.

7. Conclusions

This paper discussed the security threats of the Ethereum blockchain, the attack scenes of these threats, and their protection schemes. Ten security attacks were studied at different levels of Ethereum, which mainly included the application layer, the smart contract layer, and the network layer. The paper presented the corresponding preserved methods in detail according to their attack principles. In general, improving the quality of Ethereum smart contracts can fundamentally prevent attacks. Finally, we evaluate these protection schemes by experiments.

In the future, on the basis of studying public chain security, alliance chain security and cross-chain security will be studied, and security protection schemes for multi-attack scenarios between cross-chains will be realized. The automatic attack detection of cross-chain system is also our important research direction.

Data Availability

No Data Support.

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 2020YFB1005604, Guangxi Key Laboratory of Cryptography and Information Security (No. GCIS201915), the National Natural Science Foundation of China under Grant No. 61902021, Beijing Natural Science Foundation under Grant No. 4212008, supported in part by the National Key R&D Program of China under projects 2020YFB1006003, the Guangdong Key R&D Program under project 2020B0101090002, and the Guangxi Natural Science Foundation under grants 2018GXNSFDA281054.

References

- [1] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, vol. 37, 2021.
- [2] B. C. Ghosh, T. Bhartia, S. K. Addya, and S. Chakraborty, "Leveraging public-private blockchain interoperability for closed consortium interfacing," 2021, <https://arxiv.org/abs/2104.09801>.
- [3] S. Nakamoto, *Bitcoin: A Peer-To-Peer Electronic Cash System*, Decentralized Business Review, 2008, <https://bitcoin.org/bitcoin.pdf>.
- [4] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.
- [5] N. Szabo, *Formalizing and Securing Relationships on Public Networks*, First monday, 1997.
- [6] Y. Li, H. Liu, Z. Yang et al., "Protect your smart contract against unfair payment," in *Proceedings of the International Symposium on Reliable Distributed Systems (SRDS)*, pp. 61–70, IEEE, Shanghai, China, September 2020.
- [7] L. Luu, D. H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 254–269, Vienna, Austria, October 2016.
- [8] J. Krupp and C. Rossow, "teether: gnawing at ethereum to automatically exploit smart contracts," *27th USENIX Security Symposium (USENIX Security 18)*, pp. 1317–1333, Baltimore, MD, USA, August 2018.
- [9] P. Tsankov, A. Dan, D. Drachler-Cohen, and A. Gervais, "Securify: practical security analysis of smart contracts," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 67–82, Toronto, Canada, October 2018.
- [10] S. Kalra, S. Goel, M. Dhawan, and S. Sharma, "Zeus: analyzing safety of smart contracts," *Ndss*, pp. 1–12, San Diego, CA, USA, February 2018.
- [11] J. Gao, H. Liu, C. Liu, Q. Li, Z. Guan, and Z. Chen, "Easyflow: keep ethereum away from overflow," in *Proceedings of the IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pp. 23–26, IEEE, Montreal, QC, Canada, May 2019.
- [12] S. So, S. Hong, and H. Oh, "SMARTTEST: effectively hunting vulnerable transaction sequences in smart contracts through language model-guided symbolic execution," *30th USENIX Security Symposium (USENIX Security 21)*, August 2021.
- [13] L.-Q. Tu, X.-B. Sun, J.-L. Zhang, J. Cai, B. Li, and L.-L. Bo, "Survey of vulnerability detection tools for smart contracts," *Computer Science*, vol. 48, no. 11, pp. 79–88, 2021.
- [14] C. Hou, M. Zhou, Y. Ji, P. Daian, G. Fanti, and A. Juels, "SquirRL: automating attack discovery on blockchain incentive mechanisms with deep reinforcement learning," 2019, <https://www.arxiv-vanity.com/papers/1912.01798/>.
- [15] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, 2021.
- [16] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, vol. 36, no. 10, 2021.
- [17] Y. Marcus, E. Heilman, and S. Goldberg, "Low-resource eclipse attacks on ethereum's peer-to-peer network," *IACR Cryptology ePrint Archive*, vol. 1, no. 236, pp. 1–26, 2018.
- [18] D. Li, J. Liu, Z. Tang, Q. Wu, and Z. Guan, "AgentChain: a decentralized cross-chain exchange system," in *Proceedings of the trust security and privacy in computing and communications*, pp. 491–498, Rotorua, New Zealand, August 2019.
- [19] A. Sonnino, S. Bano, M. Al-Bassam, and G. Danezis, "Replay attacks and defenses against cross-shard consensus in sharded distributed ledgers," *IEEE, in Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 294–308, Genoa, Italy, September 2020.
- [20] A. Li, X. Wei, and Z. He, "Robust proof of Stake: a new consensus protocol for sustainable blockchain systems," *Sustainability*, vol. 12, 2020.
- [21] H. Wang, Y. Cen, and X. Li, "Blockchain router: a cross-chain communication protocol," in *Proceedings of the 6th international conference on informatics, environment, energy and applications*, pp. 94–97, Jeju Republic of Korea, March 2017.
- [22] L. Luu, J. Teutsch, R. Kulkarni, and P. Saxena, "Demystifying incentives in the consensus computer," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 706–719, Denver, CO, USA, October 2015.
- [23] L. Luu, V. Narayanan, C. Zhena, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 17–30, Vienna, Austria, October 2016.
- [24] T. D. Nguyen, L. H. Pham, and J. Sun, "SGUARD: towards fixing vulnerable smart contracts automatically," 2021, <https://arxiv.org/abs/2101.01917>.
- [25] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (sok)," in *Proceedings of the International conference on principles of security and trust*, pp. 164–186, Springer, 2017.
- [26] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on ethereum systems security: vulnerabilities, attacks, and defenses," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–43, 2020.
- [27] Y. Chen, S. Dong, T. Li, and Y. Wang, H. Zhou, "Dynamic multi-key FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5239–5249, 2021.
- [28] E. Deirmentzoglou, G. Papakyriakopoulos, and C. Patsakis, "A survey on long-range attacks for proof of stake protocols," *IEEE Access*, vol. 7, pp. 28712–28725, 2019.

- [29] Y. Huang, J. Tang, Q. Cong, and A. Lim, "Do the rich get richer? Fairness analysis for blockchain incentives," in *Proceedings of the 2021 International Conference on Management of Data*, pp. 790–803, 2021.
- [30] 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.
- [31] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts," *IACR Cryptol.ePrint Arch.* vol. 2016, 2016.
- [32] Github, "Safemath," 2021, <https://github.com/OpenZeppelin/openzeppelin-contracts-upgradeable/blob/9a42784ccc26980ca48d79191edc91e8af2185ed/contracts/utils/math/SafeMathUpgradeable.sol>.

Research Article

A Decentralized Electronic Reporting Scheme with Privacy Protection Based on Proxy Signature and Blockchain

Huiying Zou,^{1,2} Xiaofan Liu ,² Wei Ren ,^{1,2,3} and Tianqing Zhu²

¹State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China

²School of Computer Science, China University of Geosciences, Wuhan, China

³Yunnan Key Laboratory of Blockchain Application Technology, Kunming, China

Correspondence should be addressed to Wei Ren; weirencs@cug.edu.cn

Received 19 October 2021; Revised 6 January 2022; Accepted 8 January 2022; Published 7 February 2022

Academic Editor: Mamoun Alazab

Copyright © 2022 Huiying Zou 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 electronic reporting system can alleviate the problems in terms of efficiency, content confidentiality, and reporter privacy imposed in the traditional reporting system. Relying on anonymity, the privacy of reporters can be protected, but the authentication of reporters with fake names should also be maintained. If authenticated anonymity is guaranteed, the reporters may still conduct misbehaviors such as submitting fake reports after the authentication. To address the above dilemma, we propose to apply a proxy signature to achieve authenticated anonymity and employ blockchain to maintain anonymity yet guarantee traceability for reporters' misbehaviors. We also propose a new proxy signature scheme in this paper by module lattice for postquantum security. The extensive analysis justified our proposed scheme is secure and manageable.

1. Introduction

Unlike the traditional offline reporting method, where reporting letter is written by a reporter and sent to the relevant department, the electronic reporting system is more convenient and efficient. Anyone can report some content about anyone to a special department at any time anywhere. However, electronic reporting usually meets with some security problems as follows: to protect the identity of reporters, reporters usually must be anonymous. A dilemma thus arises in how to authenticate the reporter whose names are fake; if they can be authenticated in a fake name, also called authenticated anonymity, they may further report fake information. Hence, traceability should also be guaranteed.

We observe that current research have not extensively addressed the above dilemma. Or only solve one-half of the problem, either (authenticated) anonymity or traceability. In this paper, we try to solve both “birds” together with one stone. More specifically, we apply a proxy signature to achieve authenticated anonymity and we employ blockchain to obtain traceability.

Proxy signature is a kind of advanced digital signature, to which the proxy signer is delegated to generate the signature on behalf of the original signer. The reporter can send his message to a reputable third party to check, and the third-party delegate to generate the signature based on his own report message.

Blockchain presents the properties of immutability, distribution, and nonrepudiation. The reporter is a node of the blockchain, which only communicates with the trusted third party. If a malicious node wants to forge a reporting message, it is easy to find out by blockchain. According to the properties of proxy signature and blockchain, we can use them to guarantee that every reporter is honest and the reporting message is credible.

In this paper, we design this electronic reporting scheme with privacy protection based on proxy signature and blockchain. The contributions of this paper are as follows:

- (i) We apply proxy signature and blockchain technology for building a decentralized electronic reporting scheme. At the same time, our proposed scheme can achieve auditable yet authenticated anonymity, that is, preserve the reporter's privacy, authenticate

anonymous reporters upon reporting, and trace misbehaviors of anonymous reporters.

- (ii) We propose a new postquantum proxy signature based on the module lattice and provide the complete correctness analysis.

The rest of the paper is organized as follows: Related and required background information is briefly introduced in Section 2 and Section 3. In Section 4, we describe both the system models and the adversary models. We illustrate our proposed scheme in Section 5 and evaluate its security and efficiency in Section 6. Finally, we conclude this paper in Section 7.

2. Related Works

In 1996, Mambo, Usuda, and Okamoto [1, 2] proposed the idea and algorithm of proxy signature in the ACM CCS96 conference. Proxy signatures are now widely used in blockchain technology. Wang et al. [3] proposed a proxy signature mechanism based on the ElGamal algorithm in order to address the problem that the signature power of nodes cannot be transferred to blockchains, which is suitable for the management model of Sharing energy storage (SES) on blockchains. Shen et al. [4] proposed a lightweight threshold certificate authority framework LTCA by devising a threshold proxy signature, where the proxy signing key is issued by a coalition of a threshold number of certificate authorities (CAs) playing the roles of authorized nodes in the consortium blockchain. Then based on the proposed LTCA, an efficient privacy-preserving location-based service protocol (PPVC) is contrived to protect each vehicle's conditional identity privacy with a moderate cost. Pawlak et al. [5], based on the multiproxy signature technology, used the idea of a multi-intelligence system and intelligent agents and proposed a blockchain-based Internet voting system with end-to-end verifiable and auditable implementation. On the one hand, many other theoretical schemes about the proxy signature have been proposed [6–8]. On the other hand, blockchain, as a novel distributed consensus scheme, also plays a great role in various fields [9–12]. Besides, there are also other similar works [14, 15, 26, 28, 29].

In recent years, e-government has been stepping into the relationship between the government and citizens in many countries [16, 17]. It has become a powerful assistant for the government to serve the people. Among them, e-reporting has beaten traditional reporting with absolute advantages of convenience and security and has become the main way for citizens to exercise their reporting rights. The research on electronic reporting is constantly updated and improved with the development of the Internet. Wang et al. [18] first proposed the concept of a blockchain-based anonymous reporting mechanism (BB2AR), and on this basis, they proposed and implemented a BB2AR scheme based on elliptic curve public key cryptosystem. Adeshina and Ojo [19] proposed a new secure reporting system based on bit commitment. The scheme keeps the reporter's privacy in an ordinary routine, but the anonymity can be removed by a trusted thirty party (TTP) with the cooperation from the

electronic reporting center (EIC). Wang et al. [20] have come up with ReportCoin, a blockchain-based incentive anonymous reporting system that ensures the confidentiality of user identities and the reliability of reporting messages. Most of the existing electronic reporting schemes use group signature or ring signature, which are designed with the anonymity of the reporter as the necessary requirement. The related works are illustrated in Table 1.

To sum up, we combined several advantages of existing research work and designed a new electronic reporting scheme to meet the requirements of unforgeability and immutability.

3. Preliminaries

3.1. Proxy Signature. Proxy signature was first proposed by Mambo, Usuda, and Okamoto [12] in 1996. Proxy signature is a special signature scheme, in which the original signer grants his signature right to the proxy signer, and the proxy signer can generate a valid digital signature on behalf of the original signature. A proxy signature algorithm usually has the following five steps:

- (1) Initialization: generating the key and other parameters required for proxy signature according to the algorithm.
- (2) Parameter transfer: the original signer calculates the parameters that the proxy signer requires for signing and secretly transmits them to the proxy signer.
- (3) Verification of signing right: the proxy signer verifies the parameters he received. If the verification is successful, the signing process can start. If the verification fails, the original signer can be required to perform the first two steps again or the proxy signer can terminate the signing process.
- (4) Proxy signature: the proxy signer uses his or her signing power to generate a valid proxy signature for the message.
- (5) Signature verification: the party receiving the message verifies if the proxy signature is valid.

3.2. Lattice. Lattice cryptosystem is an antiquantum computing cryptosystem based on NP-hard problems. Lattice theory was initially used in cryptanalysis until Ajtai first proved the difficulty of lattice problems [21] and proposed lattice cryptography with Dwork [22].

Our scheme's security is based on the hardness of the module version of the Short Integer Solution (MSIS) and Learning With Errors problem (MLWE). The distribution of MLWE is randomly distributed a pair (a_i, b_i) from $R_q^l \times R_q$. a_i is chosen uniformly from R_q^l , and $b_i = a_i^T s + e_i$ where $e_i \leftarrow S_\eta$ and $s \leftarrow R_q$. The MLWE is commanded to recover s , while giving lots of samples from the MLWE distribution. It is stated that recovering s is impossible, though given $A \leftarrow R_q^{k \times l}$ and $b = As + e$ where $k = \text{poly}(1^\lambda)$, where λ is a secure parameter. The MSIS problem is that given β and $A \leftarrow R_q^{h \times l}$ where $h = \text{poly}(1^\lambda)$, to find a short nonzero preimage x in the lattice which satisfies $Ax = 0$ and $x \leq \beta$.

TABLE 1: The relevant related work.

Related paper	Use blockchain or not	Signature type
[3]	No	Proxy signature
[4]	Yes	Threshold signature
[5]	Yes	Multi-proxy signature
[16]	Yes	Ring signature
[17]	Yes	No signature
[19]	No	No signature
[20]	Yes	Ring signature

However, it is also impossible to find an efficient preimage x in polynomial time.

3.3. Blockchain. Blockchain development began between 2007 and 2009. It is the underlying technology of Bitcoin, known as the “public ledger for storing cryptocurrencies.” In fact, although blockchain appeared with Bitcoin, its development not only enhances the value of Bitcoin but also occupies a place for itself in the Internet field. Blockchain has many significant advantages:

Distributed storage: blockchain enables credit-based peer-to-peer transactions in distributed systems where nodes do not need to trust each other.

Immutable: the attacker’s control of a single node cannot affect the block data of other nodes and the entire network, and the cost of a successful attack is very high.

Openness: any data content and operation behavior of blockchain are publicly accessible to all nodes in the network.

4. Problem Formulation

4.1. Problem Statement. Reporting is one of the important ways for citizens to participate in politics, and it is also an important way to protect social fairness and civil rights. However, the traditional reporting way is not secure and secret for the reporter since the privacy of the reporter is easy to be exposed by going to the prosecution center or writing a reporting letter. Thus, anonymous reporting is a good way to protect reporters. It would be complicated and inconvenient if the reporting message is false since anyone can easily report without exposing their identity. To deal with this kind of reporting clutter, we can use the blockchain.

Blockchain provides the platform for everyone to join in politics with an equal chance. Users in blockchain can use the assumed name to report the bad people since blockchain has the property of anonymity. To reduce the above kind of reporting clutter, we design a reporting system using the proxy signature based on the blockchain. We randomly predetermined several proxy signers. Only the message signed from them can be verified and then be trusted by the prosecution center. Besides, considering the continuous development of quantum technology, we design a module-lattice-based proxy signature for our reporting system.

4.2. System Model. Our reporting scheme is deployed in the blockchain system. Users in the blockchain play 4 roles: reporter, proxy signer, electronic reporting box, and reporting center.

Reporting center is one special node in the blockchain system and is the trusted third party. Reporting center is voted by all users in the blockchain using the Raft algorithm [27] (Raft is a consensus algorithm for managing a replicated log). Reporting center records the reporter’s reporting signature and her/his own privacy in case of the malicious user interferes with the normal operation of the reporting system. When the user provides false reporting information, she/he will be found out by reporting center according to the ever records, and reporting center will broadcast her/his identity and remove her/him. Besides, reporting center also masters the right of permitting the electronic reporting box to verify the signature.

The electronic reporting box is predetermined by reporting center, and one reporting system only has one reporting box. The reporting box collects the reporting signatures and verifies their validity. When one user in this blockchain is reported more than half of the ordinary users (ordinary users do not contain the nodes of reporting box, reporting center, and proxy signer), the reporting box will broadcast her/his crime and remove her/him from the blockchain.

Proxy signer is the blockchain’s user whose reporting box and reporting center both trust. A complete reporting system usually has more than one proxy signer, but to explain the process of our scheme for convenience, we suppose only one proxy signer in this system. The proxy signer first authenticates the reporter’s identity and then signs for the reporting message if authentication passes.

The reporter can be any of the rest users in the blockchain and can report anyone she/he thinks is a bad guy. The reporter communicates with the proxy signer and authenticates herself/himself, and after receiving the proxy signature from the proxy signer, she/he should submit her/his privacy and signature to the reporting center.

The overall structure is illustrated in Figure 1.

4.3. Adversary Model. For the traditional reporting system, the following risks often exist:

- (1) Suppose that an adversary \mathcal{A} attacks the system, which could lead to the loss of the reporter’s privacy
- (2) The proxy signature may not be the reporter’s real proxy signature
- (3) Suppose that a malicious user \mathcal{U} who reports good people, i.e., submits a false reporting message to the proxy signer

However, our proposed scheme can avoid these risks perfectly, and we will give a detailed security analysis in Section 6.

5. Proposed Scheme

5.1. Overview. Our scheme contains four parts: system initialization, proxy reporting procedure, reporting recording, and verification.

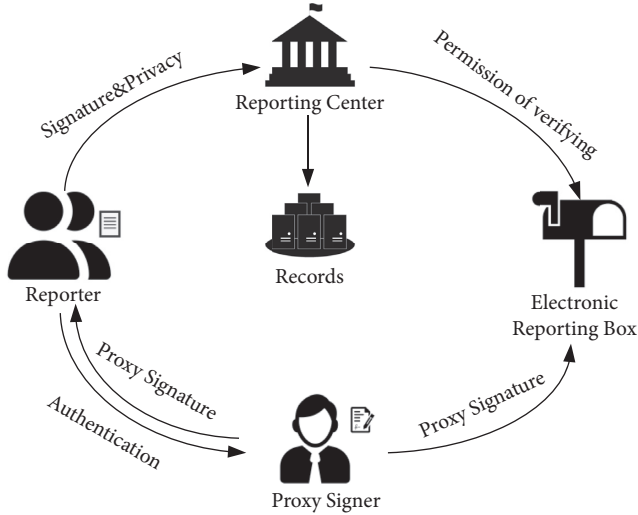


FIGURE 1: Our reporting system.

For the first step, system initialization, by taking secure parameters as input, users in this system obtain their own public keys and secret keys. In the proxy reporting procedure, a reporter from these members first selects a generally trusted proxy signer and communicates with her/him. Then, the proxy signer completes the authentication of the reporter and generates the proxy signature. The proxy signer sends the proxy signature to the electronic reporting box and the reporter afterward. After receiving the signature, the reporter encrypts her/his privacy (secret key and real name) and signature by the public key of a trusted third party, reporting center, and sends the ciphertext with the time stamp to this trusted third party as the record. The electronic reporting box records the current time after receiving the signature from the proxy signer and verifies whether this signature is valid or not. If the signature is valid, the reporting message will be recorded.

The above participants, including the reporter, the proxy signer, the electronic box, and reporting center, are all in the blockchain system such that our scheme can resist various adversary attacks. With the trusted third party participating, our scheme can trace the attacks from the malicious users while protecting the reporter's privacy (reporter is allowed to use assumed name to join the proxy signing interaction) in the reporting procedure, and the more detailed analysis is stated in the next section. Considering the future network environment and the improvement of the quantum technique, we design a new proxy signature scheme based on the module lattice.

According to the table of related work, we compare our work with these works. Our scheme uses blockchain technology to ensure that the honest reporter in our system can be protected and the malicious reporter can be traced. However, all these works cannot achieve this destination. Our scheme uses the proxy signature to achieve the electronic reporting, but works [3–5, 16, 20] use other signature types. The most important thing is that our scheme can resist the attack from the quantum adversary while no one else can.

5.2. System Initialization. Since our scheme is based on the module lattice, by taking the secure parameter 1^λ as input, the procedure first generates the system parameters, such as $\rho, \gamma_1, \gamma_2, \beta, k, l, q, \eta$, and the system functions, such as $\text{HighBits}()$ and $\text{LowBits}()$. After obtaining the necessary information, users in our scheme (including the reporter, proxy signer, the electronic reporting box, and “reporting center”) can use them to generate their public keys and secret keys. The key generation algorithm $\text{KeyGen}()$ is illustrated in Algorithm 1. It first generates a $k \times l$ matrix A , each of which is a polynomial in the ring $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. For the value of q and n , they are restricted tightly in [24]. The secret keys s and e are sampled randomly, and each coefficient of these key vectors is an element from R_q . The size of each coefficient is $\in \in [-\eta, \eta]$. According to the hard assumptions MLWE, the public key is computed as $t = As + e$. Then, users broadcast their public key pk in the blockchain. The public key and secret key can be used to encrypt/decrypt the transiting message among all users and sign/verify for the reporting message.

5.3. Proxy Signing Procedure. Suppose user i is a reporter, user j is the proxy signer, user b is the electronic reporting box, and user a is the “reporting center,” and the notations are listed in Table 2:

The proxy signing procedure contains 3 parts: identity authentication, proxy signing, and signature return and is introduced in Figure 2:

- (i) **Identity Authentication.** The reporter first randomly selects a vector y_i denoted by $S_{\gamma_1-1}^l$ where each coefficient of y_i should be less than $\gamma_1 - 1$. Then, he computes $w_i = Ay_i$ as the temporary key, and in order to be convenient and suitable for the next steps, he uses the function $\text{HighBits}()$ to extract the high-order bits of w_i , named as w_{i1} . w_{i1} should satisfy the equation $w_i = w_{i1} \cdot 2\gamma_2 + w_{i0}$ where $w_{i0} \leq \gamma_2$. The reporter hashes the value of w_{i1} as the challenge c_i which consists of 60 's $\{-1, 0, 1\}^*$. For the size of the challenge, consider that c_i at most contains 60 's ± 1 . To make a complete identified authentication, the reporter should “mix” the challenge c_i with her/his own secret key s_i . However, since $s_i \leq \eta$, the size of $c_i s_i$ is less than 60η . β is the maximum coefficient value of $c_i s_i$. Thus, the above condition can be written as $\beta \leq 60\eta$. The authentication requirement is $u_i = y_i + c_i s_i$, but u_i has the limited range of size where $u_i \leq \gamma_1 - \beta$. Besides, to achieve the following authentication, another limitation should be admitted, i.e., the low-order bits of $Ay_i - c_i s_i$'s coefficients should be less than $\gamma_2 - \beta$; otherwise, it will leak the information of the secret key. If the size check passed, the reporter sends (u_i, c_i, M) to the proxy signer. After receiving these information, the proxy signer identifies the reporter by using the function $\text{HighBits}()$. If the reporter's identity is confirmed, the proxy signing will be carried out next.

Procedure KeyGen()	
(1)	$A \leftarrow R_q^{k \times l}$
(2)	$(s, e) \leftarrow S_\eta^s \times S_\eta^e$
(3)	$t = As + e$
(4)	Return $(pk = (A, t), sk = (s, e))$

ALGORITHM 1: Key Generation.

TABLE 2: Notations.

Notation	Meaning
sk_i	sk_i is the reporter's secret key, $sk_i = (s_i, e_i)$
pk_i	pk_i is the reporter's public key, $pk_i = t_i$
sk_j	sk_j is the proxy signer's secret key, $sk_j = (s_j, e_j)$
pk_j	pk_j is the proxy signer's public key, $pk_j = t_j$
sk_b	sk_b is the electronic reporting box's secret key, $sk_b = (s_b, e_b)$
pk_b	pk_b is the proxy signer's public key, $pk_b = t_b$
sk_a	sk_a is the reporting center's secret key, $sk_a = (s_a, e_a)$
pk_a	pk_a is the reporting center's secret key, $pk_a = t_a$
M	M is the signing message
σ	σ is the proxy signature

(ii) *Proxy Signing.* The proxy signer makes u_i as y_j to participate in the following signing procedure. Similar as the above process, the proxy signer computes $w_j = Ay_j$ as the signing temporary key and takes the high-order bits w_{j1} of w_j . c_j is hashed from w_{j1} and the signing message M . Since the hash function of the signing procedure is the same as the identity authentication's, the size of $c_j s_j$ is also less than 60η , and the maximum coefficient value of $c_j s_j$ also is written as β where $\beta \leq 60\eta$. Thus, the potential signature z_j is constructed by $z_j = y_j + c_j s_j$. In order to protect the secret key and make the signature independent of the secret key, $z_j \leq \gamma_1 - \beta$ and also $LowBits(Ay_i - c_j e_j, 2\gamma_2)_\infty \leq \gamma_2 - \beta$ which confirms that the signature can be verified validity.

(iii) *Signature Return.* After z_j passes the size check, the proxy signer obtains the proxy signature $\sigma = (z_j, c_j, c_i)$ and sends it to the reporter and the electronic reporting box.

It is stated that to protect privacy, the information should be encrypted by using the destination's public key during the interaction.

5.4. Reporting Record. After receiving the proxy signature, to record this reporting behavior in case of malicious reporting (since the reporter is able to use the assumed name to accomplish the reporting), the reporter should send her/his own secret key sk_i and her/his real name with the signature σ to the trusted third party, named as reporting center. The reporting center stores the information secretly and only broadcasts malicious user's real identity if he tells lies in reporting procedure.

Besides, the electronic reporting box receives the signature and matches it to the previously broadcast public key.

The reporting box records the signature with its corresponding public key and waits for permission to verify the reporting center. If the reporting box has not received permission to verify for a long time (The time is set according to the blockchain latency), he will abandon this signature and mark this proxy signer. If the amount of marked users is over the half users of this system, this proxy signer will be broadcast as a malicious user and removed.

5.5. Verification. The electronic reporting box first communicates with the reporting center to confirm whether this signature has been registered or not. The verification is operated by the reporting box after getting permission from the reporting center and is illustrated in Algorithm 2. The reporting box first checks the size of z_j and verifies whether the signature is changed or not during the transmission. According to $Az_j - c_j t_j = Ay_j - c_j e_j$ and $w_j = Ay_j$, it can be written as follows.

$$Az_j - c_j t_j = w_j - c_j e_j. \quad (1)$$

Thus, it is clear that

$$HighBits(Az_j - c_j t_j) = HighBits(w_j - c_j e_j). \quad (2)$$

Because $LowBits(Ay_j - c_j e_j, 2\gamma_2)_\infty \leq \gamma_2 - \beta$ and the coefficients of $c_j e_j$ are less than β , adding other low-order coefficients cannot cause a big effect in high-order bits. Therefore, the above equations can be written as follows:

$$HighBits(Az_j - c_j t_j) = HighBits(w_j - c_j e_j) = HighBits(w_j). \quad (3)$$

If the hash value of $HighBits(Az_j - c_j t_j)$ and the signing message is equal to signature's c_j , the signature is not changed during the transmission and is verified validity. Up to here, the reporting box has verified that the signature is generated by the proxy signer and will verify whether the real signer of the signature is the reporter or not.

The verifying process is similar to the above. The reporting box writes u_i as the result of $Az_j - c_j t_j$. In the function $HighBits()$, $c_j e_j$ cannot affect the result of the computation. u_i can be approximately seen as Ay_j , in other words, Au_i . Therefore, use the reporter's public key t_i to identify who the real signer is. Since $u_i - c_i t_i = Au_i - c_i t_i$ and refer to the above equations, it is clear that

$$\begin{aligned} HighBits(u_i - c_i t_i) &= HighBits(Au_i - c_i t_i) \\ &= HighBits(Ay_i - c_i e_i). \end{aligned} \quad (4)$$

Besides, since the coefficients of $c_i e_i$ are less than β , adding other low-order bits cannot influence the high-order's. According to the above analysis, if $H(HighBits(u_i - c_i t_i)) = c_i$, it can prove that the reporter is the real signer, and the signing message can be accepted by the reporting box while the one-time proxy reporting procedure ends up.

The electronic reporting box verifies the proxy signatures from the proxy signer and collects the reporting message if signatures are valid. For the person who is reported, suppose

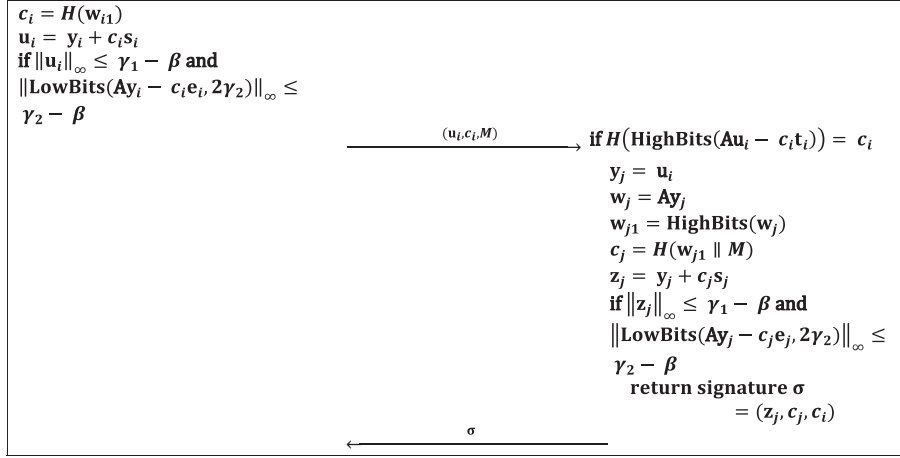
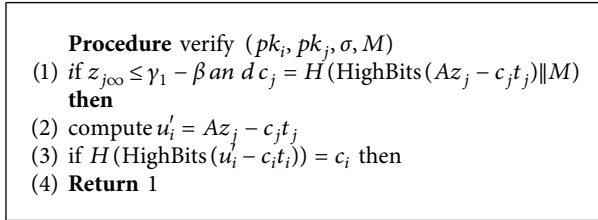


FIGURE 2: Proxy signing procedure.



ALGORITHM 2: Verification.

that she/he is user m , she/he will not be removed from this blockchain system right away. Only when the amount of signing message is over the half of the blockchain system users, the reporting center will broadcast the message “User m is the traitor, do not trust her/him” and remove user m right away.

6. Security Analysis

According to the adversary model, our scheme can resist these risks:

- (1) Suppose an adversary \mathcal{A} who wants to steal the privacy of the reporter. Since the reporter should send reporting center her/his privacy with the signature to register herself/himself, \mathcal{A} wants to steal some information from the transmission. However, our scheme state that any transiting message should be encrypted by using receiver’s public key, and the public key is generated based on the hard assumption of MLWE while the encryption in our scheme is Crystals-Kyber [25], one of the Round 3 NIST public-key encryption submissions (Because the main idea of our work is the reporting system designing, the encryption process is omitted). The above encrypted algorithm has postquantum property. Although \mathcal{A} can intercept the ciphertext, she/he is not able to obtain the real message without the reporting center’s private key or using modern technology. For the reporting center, she/he is the trusted third party, and only she/he can have access

to visit the records of reporter’s privacy so that \mathcal{A} cannot get the reporter’s privacy there. Thus, our scheme can avoid the risk of reporter’s privacy leakage.

- (2) Another risk is that the signature misses the required authentication, which means that the signature may not be the reporter’s real proxy signature.

Suppose that the proxy signer is malicious, she/he sends a false signature and claims that the signature is entrusted by the reporter, i.e., she/he frame the reporter. Because of the procedure of report recording, our scheme can prevent this risk. In our scheme, the proxy signer should also send the signature back to the reporter so that the reporter will not get the signature if she/he has not submitted the requirement of reporting signature to the proxy signer. Thus, when the proxy signer sends the signature to the electronic reporting box, the framed user will not send her/his privacy information to the reporting center such that the reporting center will not send the permission of verifying to the electronic reporting box and the verifying process will not start. If reporting box does not receive permission to verify for a long time, she/he will mark the proxy signer. When the amount of this proxy signer’s marks is over the established domain (here, we set the domain value as half of the system’s users), this proxy signer will be removed.

Another case is the user impersonates others to communicate with the proxy signer. However, it is impossible since strict identity authentication is implemented during the interaction and the user cannot obtain other’s secret key.

Therefore, our scheme can prevent users from being framed.

- (3) Suppose a malicious user \mathcal{U} submits a false reporting message to the proxy signer. Although the false reporting message can finally be signed by the proxy signer, people reported will be removed only when the amount of reporting messages from different

users is over no less than half of all users. Besides, once the reporting message needs to be broadcast after the signature is verified validity, other people will know who has been reported and they can dispute this message with the reporting center if people reported are not bad. If half of the users raise disputes for this reporting message, reporting center will search the signature records to find out the reporter. The reporter can use the assumed name to report others, but she/he has to send his private information (including her/his real name) to reporting center so that reporting center can trace her/his identity and broadcast it. That is, our scheme can find out the malicious user.

7. Conclusions

In this paper, we propose a decentralized electronic reporting scheme based on proxy signature and blockchain and provide the system model of our scheme. To resist the future quantum attack, we propose a new proxy signature based on the module lattice. While preserving the reporter's privacy, our scheme can trace the malicious users at the same time, which greatly improves the usability of our scheme. Besides, we give a detailed security analysis for the adversary model. In the future, we will improve our proposed system efficiency and make the comparison with other electronic reporting systems. [13–15, 23].

Data Availability

The signature data and the code 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 research was financially supported by the Foundation of Yunnan Key Laboratory of Blockchain Application Technology (Nos. 202105AG070005 and YNB202103), the National Natural Science Foundation of China (No. 61972366), the Provincial Key Research and Development Program of Hubei (No. 2020BAB105), and the Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No. 2019BDKFJ003 and 2019BDKFJ011).

References

- [1] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, pp. 48–57, New Delhi, India, January 1996.
- [2] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: Delegation of the power to sign messages," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 79, no. 9, pp. 1338–1354, 1996.
- [3] Y. Wang, W. Qiu, L. Dong et al., "Proxy signature-based management model of sharing energy storage in blockchain environment," *Applied Sciences*, vol. 10, no. 21, p. 7502, 2020.
- [4] H. Shen, J. Zhou, Z. Cao, X. Dong, and K.-K. R. Choo, "Blockchain-based lightweight certificate authority for efficient privacy-preserving location-based service in vehicular social networks," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6610–6622, 2020.
- [5] M. Pawlak, A. Poniszewska-Marañda, and J. Guziur, "Intelligent agents in a blockchain-based electronic voting system," in *Proceedings of the Intelligent Data Engineering and Automated Learning – IDEAL 2018. IDEAL 2018. Lecture Notes in Computer Science*, vol. 11314, 2018.
- [6] X. Jia, H. Yupu, and J. Mingming, "Lattice-based forward secure proxy signatures," *Journal of Computer Research and Development*, vol. 58, no. 3, p. 583, 2021.
- [7] R. Gao and J. Zeng, "Forward secure certificateless proxy multi-signature scheme," *International Journal of Electronic Security and Digital Forensics*, vol. 13, no. 1, pp. 1–27, 2021.
- [8] R. Huang, Z. Huang, and Q. Chen, "A generic conversion from proxy signatures to certificate-based signatures," *Journal of Internet Technology*, vol. 22, no. 1, pp. 209–217, 2021.
- [9] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "Blockchain: a distributed solution to automotive security and privacy," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, 2017.
- [10] E. Bellini, Y. Iraqi, and E. Damiani, "Blockchain-based distributed trust and reputation management systems: a survey," *IEEE Access*, vol. 8, pp. 21 127–21 151, 2020.
- [11] S. Yu, K. Lv, Z. Shao, Y. Guo, J. Zou, and B. Zhang, "A high performance blockchain platform for intelligent devices," in *Proceedings of the 2018 1st IEEE international conference on hot information-centric networking (HotICN)*, pp. 260–261, IEEE, Shenzhen, China, August 2018.
- [12] W. Liang, Y. Fan, K.-C. Li, D. Zhang, and J.-L. Gaudiot, "Secure data storage and recovery in industrial blockchain network environments," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6543–6552, 2020.
- [13] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for iiot devices," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.
- [14] H. Xiong, C. Jin, M. Alazab et al., "On the design of blockchain-based ecdsa with fault-tolerant batch verification protocol for blockchain-enabled iomt," *IEEE Journal of Biomedical and Health Informatics*, p. 1, 2021.
- [15] T. R. Gadekallu, Q. V. Pham, D. C. Nguyen et al., "Blockchain for edge of things: Applications, opportunities, and challenges," *IEEE Internet of Things Journal*, vol. 9, no. 2, p. 1, 2021.
- [16] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, vol. 7, no. 2, pp. 1204–1221, 2022.
- [17] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, 2021.
- [18] B. Wang, J. Sun, Y. He, D. Pang, and N. Lu, "Large-scale election based on blockchain," *Procedia Computer Science*, vol. 129, pp. 234–237, 2018.
- [19] S. A. Adeshina and A. Ojo, "Maintaining voting integrity using blockchain," in *Proceedings of the 2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*, pp. 1–5, IEEE, Abuja, Nigeria, December 2019.

- [20] H. Wang, D. He, Z. Liu, and R. Guo, "Blockchain-based anonymous reporting scheme with anonymous rewarding," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1514–1524, Nov. 2020.
- [21] W. Qiu, B. Liu, and S. Shi, "An impeaching system based on bit commitment with revocable anonymity," in *Proceedings of the 2010 International Conference on Internet Technology and Applications*, pp. 1–6, IEEE, Wuhan, China, August 2010.
- [22] S. Zou, J. Xi, S. Wang, Y. Lu, and G. Xu, "Reportcoin: a novel blockchain-based incentive anonymous reporting system," *IEEE Access*, vol. 7, pp. 65544–65559, 2019.
- [23] M. Ajtai, "Generating hard instances of lattice problems," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, pp. 99–108, Pennsylvania, Philadelphia, USA, July 1996.
- [24] M. Ajtai and C. Dwork, "A public-key cryptosystem with worst-case/average-case equivalence," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pp. 284–293, Texas, El Paso, USA, May 1997.
- [25] 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, USENIX Association, Philadelphia, PA, June 2014, <https://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro>.
- [26] L. Ducas, E. Kiltz, T. Lepoint et al., "CRYSTALS-dilithium: a lattice-based digital signature scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 1, pp. 238–268, 2018.
- [27] J. Bos, L. Ducas, E. Kiltz et al., "Crystals - kyber: a cca-secure module-lattice-based kem," in *Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS P)*, pp. 353–367, London, UK, April 2018.
- [28] T. R. Gadekallu, Q. V. Pham, D. C. Nguyen et al., "Blockchain for edge of things: Applications, opportunities, and challenges," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 964–988, 2021.
- [29] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *Designs, Codes and Cryptography*, vol. 75, no. 3, pp. 565–599, 2015.

Research Article

A Hybrid Design of Linkable Ring Signature Scheme with Stealth Addresses

Weizhou Li,¹ Zhiqiang Lin ,² and Qi Chen³

¹School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China

²School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China

³Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou 510006, China

Correspondence should be addressed to Zhiqiang Lin; linzhiqiang0824@163.com

Received 13 November 2021; Revised 14 December 2021; Accepted 10 January 2022; Published 7 February 2022

Academic Editor: Yuling Chen

Copyright © 2022 Weizhou Li 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 a transformational technology which affects finance, Internet, and politics. However, many privacy protection problems for blockchain are waiting to be solved. In this study, we propose a novel linkable ring signature scheme with stealth addresses, which enables the payer and payee of the transaction to be anonymous and unlinkable in the cryptocurrency. The scheme is combined with an elliptic curve discrete logarithm (ECD logarithm)-based key encapsulation mechanism (KEM) stage and a lattice-based signature stage. The master public key and master secret key are much smaller compared with the previous scheme. Complete secure proof of the scheme is also presented in this study.

1. Introduction

1.1. Background. As a novel technology, blockchain technology has been widely used in many fields since its introduction in [1–6]. The development of blockchain is also inseparable from digital signatures. Digital signature provides security and authentication for information during the process of information dissemination, such as protecting user's privacy and preventing double spending with the support of the anonymity and linkability of the ring signature scheme [7, 8].

In [9], the authors proposed a linkable ring signature scheme with stealth addresses denoted by SALRS. This scheme enables the payer and payee of the transaction to be anonymous and unlinkable in the cryptocurrency. Specifically, the linkable ring signature and stealth address [10–12] are employed in CryptoNote [10]. When a payer A wants to pay a payee B through a transaction, the payer B uses a stealth address to generate a derived public key. Then, the payer A uses the derived public key as the address of the payee B . Also, transactions cannot be identified because of the absence of the master public key. When the payee B , as a payer in transaction, wants to spend his coins on the derived

public key, he generates a linkable ring signature with the support of a set of derived public keys. In order to verify the linkable ring signature, it is not necessary for anyone to find out that the actual signer is corresponding to the derived public key. When it comes to the linkability which can prevent double spending in a transaction, if two signatures are generated by the payee B corresponding to a derived public key, they will be detected as linked because the coin corresponding to the derived public key can be used only once. In this study, we focus on concrete construction of the SALRS scheme in order to enable both payer and payee of a transaction to be hidden in the cryptocurrency.

1.2. Our Contribution. We propose a novel concrete linkable ring signature scheme with stealth addresses based on the elliptic curve discrete logarithm (ECD logarithm) for the key encapsulation mechanism (KEM) stage and lattice for signature stage. The ECD-based KEM provides smaller keys. In particular, the size of the master public key is 510 bits, and the size of the master secret key is 512 bits, which is much smaller than the ones in the previous scheme in [9]. Moreover, all the secure properties which a SALRS should

have, including unforgeability, linkability, nonslanderability, anonymity, master-public-key-unlinkability, and derived-public-key-unlinkability, still keep.

1.3. Organization of This Paper. The rest of the study is organized as follows. Preliminaries are given in Section 2. In Section 3, we formally propose the SALRS scheme. Afterwards, the security models of the SALRS scheme are presented in Section 4. As a vital content of this study, in Section 5, our concrete SALRS scheme is showed. In Section 6, we analyze and prove the security of our SALRS scheme. Moreover, efficiency analysis, especially less storage cost of our SALRS scheme, is introduced in Section 7. Finally, we summarize this study and come out conclusions in Section 8.

1.4. Related Work. There are a lot of classic linkable ring signature schemes relied on the hardness number-theoretic problems, such as [13, 14]. Many of them have specific application scenarios, for example, [15, 16] are based on certificates and identity-based, respectively. However, a lot of cryptographic schemes based on classical number theories are suffered from future quantum computer's threats [17]. All the same, some advantages of cryptographic schemes relying on classical number theory, for example, the elliptic curve discrete logarithm [18], are faster calculation and less storage cost.

Lattice-based ring signatures were first introduced by Brakerski and Tauman-Kalai in 2010. They proposed a construction of ring signature scheme based on SIS assumption. Then, in 2013, Melchor et al. proposed a ring signature scheme based on LWE assumption. Until now, many lattice-based ring signature schemes have been proposed, such as [19–21].

The existing works on the linkable ring signature and stealth address have been proposed, e.g., [8, 22]. However, most of the existing works above either merely consider linkable ring signature or stealth address rather than both of them. Fortunately, literature [9] has successfully proposed a new cryptographic primitive denoted by SALRS. The new cryptographic primitive has not only combined the linkable ring signature with the stealth address but also captured adversarially chosen key attacks in the linkability model. Additionally, it is also potentially quantum resistant.

2. Preliminaries

In this section, before showing our concrete SALRS construction, we give some preliminary results about the mathematical background concerning bilinear maps and lattice and complexity assumptions. For more details, please refer to [23–26].

2.1. Mathematical Background. Let $\mathbb{G}_1, \mathbb{G}_2,$ and \mathbb{G}_T be the groups of prime order p , and g_i be a generator of \mathbb{G}_i ($i = 1, 2$). Set $\mathbb{G}_1 \neq \mathbb{G}_2$, and there exists no efficient homomorphism between \mathbb{G}_1 and \mathbb{G}_2 . We say that $e: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a

bilinear, efficient, and computable map if it satisfies the following two properties.

- (1) Bilinear: for $\forall a, b \in \mathbb{Z}_p$, where the integers modulo p is denoted by \mathbb{Z}_p , we have $e(g_1^a, g_1^b) = e(g_1, g_1)^{ab}$.
- (2) Efficient: $e(g_1, g_2) \neq 1$.

Let q and n be two positive integers, and denote the integers modulo q by \mathbb{Z}_q , which will be represented in the range $-(q/2), (q/2]$ or $[-(q-1/2), (q-1/2)]$, where q is even or odd, respectively. Let R and R_q be the rings $\mathbb{Z}[X]/(X^n+1)$ and $\mathbb{Z}_q[X]/(X^n+1)$, respectively. We set $r = a_0 + a_1X + \dots + a_{n-1}X^{n-1} \in R$ and $\mathbf{r} = (r_1, \dots, r_k) \in R^k$ to define the $l_1, l_2,$ and l_∞ norms of r and \mathbf{r} as follows:

- (1) $\|r\|_\infty \triangleq \max |a_i|$
- (2) $\|r\|_1 \triangleq \sum_i |a_i|$
- (3) $\|r\|_2 \triangleq \sqrt{|a_0|^2 + \dots + |a_{n-1}|^2}$
- (4) $\|\mathbf{r}\|_\infty \triangleq \max \|r_i\|_\infty$
- (5) $\|\mathbf{r}\|_1 \triangleq \sum \|r_i\|_1$
- (6) $\|\mathbf{r}\|_2 \triangleq \sqrt{\|r_1\|_2^2 + \dots + \|r_k\|_2^2}$

We also denote two sets:

- (1) $\text{SS}_\eta \triangleq \{r \in R \mid \|r\|_\infty \leq \eta\} \setminus \{r \in R \mid \|r\|_\infty \leq \eta\}$
- (2) $\mathbf{B}_\theta \triangleq \{r \in R_q \mid r \text{ has } \theta \text{ coefficients that are } \pm 1 \text{ and the rest are } 0\}$

2.2. Complexity Assumptions. The security of our novel SALRS scheme is based on bilinear Diffie-Hellman 1 assumption, module-SIS assumption, and module-LWE assumption.

2.2.1. Bilinear Diffie-Hellman 1 (BDH-1) Assumption. Let $\mathbb{G}_1, \mathbb{G}_2,$ and \mathbb{G}_T be groups of prime order p , and g_i be a generator of \mathbb{G}_i ($i = 1, 2$). Set $\mathbb{G}_1 \neq \mathbb{G}_2$, and there exists no efficient homomorphism between \mathbb{G}_1 and \mathbb{G}_2 . $e: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a bilinear, efficient, and computable map. The Bilinear Diffie-Hellman 1 (BDH-1) problem is that given g_1, g_2, g_1^a, g_1^b , compute $e(g_1, g_2)$, where $a, b \in \mathbb{Z}_p$.

2.2.2. Module-SIS Assumption. The module-SIS problem with parameters (n, q, k, l, β) is that for uniformly random $\mathbf{A} \in R_q^{k \times l}, \mathbf{t} \in R_q^k$, and $k \times k$ identity matrix \mathbf{I} , find $\mathbf{x} \in R_q^{k+l}$, such that $\|\mathbf{x}\|_2 \leq \beta$ and $[\mathbf{A} \mid \mathbf{I}] \cdot \mathbf{x} = \mathbf{t}$. The problem can be adapted into the infinity-norm version, where $\|\mathbf{x}\|_\infty \leq \beta$. Additionally, the homogeneous version of the module-SIS problem is defined with $\mathbf{t} = \mathbf{0}$ and $\mathbf{x} \neq \mathbf{0}$.

2.2.3. Module-LWE Assumption. The module-LWE problem with parameters (n, q, k, l, η) is that for uniformly random $\mathbf{A} \in R_q^{k \times l}$, let $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \in R_q^k$, where $\mathbf{s} \in S_\eta^l, \mathbf{e} \in S_\eta^k$ have their entries selected concerning some distributions (uniform distribution, Gaussian distribution) over S_η . There are two versions about module-LWE. The search variant of module-LWE is to find \mathbf{s} given (\mathbf{A}, \mathbf{b}) . The decision variant is to distinguish (\mathbf{A}, \mathbf{b}) from a uniformly random pair over

$R_q^{k \times l} \times R_q^k$. In this study, we use a transformed version of the decision variant of module-LWE, which is to distinguish $(\mathbf{A}, \mathbf{As})$ from (\mathbf{A}, \mathbf{r}) , where $\mathbf{A} \leftarrow R_q^{k \times l}$, $\mathbf{s} \leftarrow S_\eta^l$, and $\mathbf{r} \leftarrow R_q^k$.

3. SALRS Scheme

The syntax of linkable ring signature scheme with stealth addresses (SALRS) was first purposed by [9], which realizes the cryptographic functions that a cryptocurrency wants to hide payers and payees of transactions. There are eight algorithms in a SALRS scheme.

Setup $(\lambda) \rightarrow \text{PP}$: the input to this algorithm is a security parameter λ and outputs the public parameters PP.

MasterKeyGen (PP) $\rightarrow (\text{MPK}, \text{MSK})$: the input to this algorithm is the public parameters (PP) and outputs the user's master key pair (MPK, MSK) (master public key, master secret key).

DerivedPublicKeyGen (MPK) $\rightarrow \text{DPK}$: the input to this algorithm is a master public key (MPK) and outputs the derived public key (DPK).

DerivedPublicKeyOwnerCheck (DPK, MPK, MSK) $\rightarrow 1/0$: the input to this algorithm is a derived public key (DPK) and a master key pair (MPK, MSK) and outputs $b \in \{0, 1\}$. 1 and 0 indicate that the derived public key (DPK) is valid or invalid, respectively.

DerivedPublicKeyPublicCheck (DPK) $\rightarrow 1/0$: the input to this algorithm is a derived public key (DPK) and outputs $b \in \{0, 1\}$. 1 and 0 indicate that the derived public key (DPK) is well-formed or not well-formed, respectively.

Sign $(M, R, \text{DPK}, \text{MPK}, \text{MSK}) \rightarrow \sigma$: the input to this algorithm is a message M , a ring of well-formed derived public keys $R = (\text{DPK}_1, \dots, \text{DPK}_r)$ (where we regard the public key ring R as an order set, namely, it consists of the public keys which are ordered and have indexes), a derived public key $\text{DPK} \in R$, and a master key pair (MPK, MSK) for the derived public key (DPK) and outputs a signature σ .

Verify $(M, R, \sigma) \rightarrow 1/0$: the input to this algorithm is a message M , a ring of well-formed derived public keys R , and a signature σ and outputs $b \in \{0, 1\}$. 1 and 0 indicate that the signature σ is valid or invalid, respectively.

Link $(M_0, R_0, \sigma_0, M_1, R_1, \sigma_1) \rightarrow 1/0$: the input to this algorithm is two valid (message M , derived public key ring R , signature σ) tuples (M_0, R_0, σ_0) and (M_1, R_1, σ_1) and outputs 0 or 1. 1 and 0 indicate that the two signatures are linked or unlinked, respectively.

4. Security Model of SALRS

A SALRS scheme should be correctness, unforgeable, linkable, non-slanderable, anonymous, master-public-key-unlinkable, and derived-public-key-unlinkable, which ensure the scheme satisfying the security and privacy protection requirements of cryptocurrencies in most practical settings.

In the following games, we use \mathcal{A} or $n()$ to denote any probabilistic polynomial time (PPT) adversary or polynomial, respectively.

4.1. Correctness. Correctness means that one can derive a "right" feedback while honestly performing the protocols.

Let $\text{PP} \leftarrow \text{setup}(\lambda)$,

- (1) For any $(\text{MPK}, \text{MSK}) \leftarrow \text{MasterKeyGen}(\text{PP})$ and any $\text{DPK} \leftarrow \text{DerivedPublicKeyGen}(\text{MPK})$, we have $\text{DerivedPublicKeyOwnerCheck}(\text{DPK}, \text{MPK}, \text{MSK}) = 1$ and $\text{DerivedPublicKeyPublicCheck}(\text{DPK}) = 1$.
- (2) For any message M , any ring of well-formed derived public keys R , and any derived public key $\text{DPK} \in R$, such that $\text{DerivedPublicKeyOwnerCheck}(\text{DPK}, \text{MPK}, \text{MSK}) = 1$ for some master key pair (MPK, MSK) , we have $\text{verify}(M, R, \text{sign}(M, R, \text{DPK}, \text{MPK}, \text{MSK})) = 1$.
- (3) For any message M_i , any ring of well-formed derived public keys R_i , and any derived public key $\text{DPK}_i \in R_i$, such that $\text{DerivedPublicKeyOwnerCheck}(\text{DPK}_i, \text{MPK}_i, \text{MSK}_i) = 1$ for some master key pair $(\text{MPK}_i, \text{MSK}_i)$, let $\sigma_i \leftarrow \text{sign}(M_i, R_i, \text{DPK}_i, \text{MPK}_i, \text{MSK}_i)$ ($i = 0, 1$). We have $\text{link}(M_0, R_0, \sigma_0, M_1, R_1, \sigma_1) = 1$ if $\text{DPK}_0 = \text{DPK}_1$, and $\Pr[\text{link}(M_0, R_0, \sigma_0, M_1, R_1, \sigma_1) = 0] \geq 1 - \text{negl}(\lambda)$ if $\text{DPK}_0 \neq \text{DPK}_1$, where negl is a negligible function.

4.2. Unforgeability. Unforgeability means that only the user who knows the secret key for some public key in a ring can generate a valid signature.

4.2.1. Setup. $\text{PP} \leftarrow \text{setup}(\lambda)$ is run. PP is given to \mathcal{A} . $\{(\text{MPK}_i, \text{MSK}_i) \leftarrow \text{MasterKeyGen}(\text{PP})\}_{i=1}^{n(\lambda)}$ are run and $\{\text{MPK}_i\}$ are given to \mathcal{A} .

4.2.2. Probing Phase. \mathcal{A} can query the following oracles:

- (1) Derived Public Key Adding Oracle, $\text{ODPKAdd}()$: it means that $\text{ODPKAdd}(\text{DPK}, \text{MPK})$ returns $b \leftarrow \text{DerivedPublicKeyOwnerCheck}(\text{DPK}, \text{MPK}, \text{MSK})$ to \mathcal{A} . If $b = 1$, set $L_{dpk} = L_{dpk} \cup \{\text{DPK}\}$, where $L_{dpk} = \emptyset$ is initialized.
- (2) Signing Oracle, $\text{OSign}()$: it means that $\text{OSign}(M, R, \text{DPK})$, where DPK

$\in R \cap L_{dpk}$, returns $\sigma \leftarrow \text{sign}(M, R, \text{DPK}, \text{MPK}, \text{MSK})$ to \mathcal{A} , where (MPK, MSK) is the master key pair for DPK .

4.2.3. Output Phase. \mathcal{A} outputs a message M^* , a ring of well-formed derived public keys R^* , and a signature σ^* .

Let $S_{so} = \{(M, R, \text{DPK}, \sigma)\}$ be the query-answer tuples for OSign . \mathcal{A} succeeds if

- (1) $\text{Verify}(M^*, R^*, \sigma^*) = 1$ and
- (2) $R^* \subseteq L_{dpk}$ and
- (3) $(M^*, R^*, \sigma^*) \notin S_{so}$, where $?$ means that (M^*, R^*, σ^*) is not a tuple obtained by querying OSign .

Definition 1. The SALRS is unforgeable if for all \mathcal{A} , $A dv_{\mathcal{A}}^{uf}$ is negligible, where $A dv_{\mathcal{A}}^{uf} = \Pr[\mathcal{A} \text{ succeeds}]$. We name the game for unforgeability Game_{uf} .

4.3. Linkability. Linkability means that if the key owner generates two or multiple valid signatures with respect to one derived public key, the signatures will be found to be linked.

4.3.1. Setup. $\text{PP} \leftarrow \text{setup}(\lambda)$ is run. PP is given to \mathcal{A} .

4.3.2. Output Phase. \mathcal{A} outputs k ($k \geq 2$) tuples $(M_i^*, R_i^*, \sigma_i^*)$ ($i = 1, \dots, k$).

\mathcal{A} succeeds if

- (1) $\text{Verify}(M_i^*, R_i^*, \sigma_i^*) = 1$ and
- (2) $\text{Link}(M_i^*, R_i^*, \sigma_i^*, M_j^*, R_j^*, \sigma_j^*) = 0$ ($\forall i, j \in [1, k], s.t. i \neq j$) and
- (3) $|\cup_{i=1}^k R_i^*| < k$.

Definition 2. The SALRS is linkable if for all \mathcal{A} , $A dv_{\mathcal{A}}^{\text{link}}$ is negligible, where $A dv_{\mathcal{A}}^{\text{link}} = \Pr[\mathcal{A} \text{ succeeds}]$. We name the game for linkability $\text{Game}_{\text{link}}$.

4.4. Nonslanderability. Nonslanderability means that no one can frame other users by creating a signature which is linked to a signature of the target user.

4.4.1. Setup. Same as that of Game_{uf} .

4.4.2. Probing Phase. Same as that of Game_{uf} .

4.4.3. Output Phase. \mathcal{A} outputs two tuples (M', R', σ') and (M^*, R^*, σ^*) .

Let $S_{so} = \{(M, R, \text{DPK}, \sigma)\}$ be the query-answer tuples for OSign. \mathcal{A} succeeds if

- (1) $\text{Verify}(M^*, R^*, \sigma^*) = 1$ and
- (2) $(M', R', \sigma') \in S_{so}$ for some derived public keys $\text{DPK}' \in R', \cap L_{dpk}$ and
- (3) $(M^*, R^*, \text{DPK}', \sigma^*) \notin S_{so}$ and
- (4) $\text{Link}(M^*, R^*, \sigma^*, M', R', \sigma') = 1$

Definition 3. The SALRS is nonslanderable if for all \mathcal{A} , $A dv_{\mathcal{A}}^{ns}$ is negligible, where $A dv_{\mathcal{A}}^{ns} = \Pr[\mathcal{A} \text{ succeeds}]$. We name the game for nonslanderability Game_{ns} .

4.5. Anonymity. Anonymity means that no one can identify the signer's derived public key out of the ring, with a valid signature with respect to a ring of derived public keys.

4.5.1. Setup. Same as that of Game_{uf} .

4.5.2. Probing Phase 1. Same as the probing phase of Game_{uf} .

4.5.3. Challenge Phase. \mathcal{A} outputs a message M^* , a ring of well-formed derived public keys R^* , and two distinct indices $1 \leq i_0, i_1 \leq n(\lambda)$, such that

- (1) $\text{DPK}_{i_0}, \text{DPK}_{i_1} \in R^* \cap L_{dpk}$
- (2) None of OSign with DPK_{i_0} and DPK_{i_1} was queried.

A random bit $b \in \{0, 1\}$ is chosen, and \mathcal{A} is given the $\sigma \leftarrow \text{sign}(M^*, R^*, \text{DPK}_{i_b}, \text{MPK}, \text{MSK})$, where (MPK, MSK) is the master key pair for DPK_{i_b} .

4.5.4. Probing Phase 2. Same as the probing phase 1, but with the restriction that OSign with DPK_{i_0} and DPK_{i_1} cannot be queried.

4.5.5. Output Phase. \mathcal{A} outputs a bit b' as its guess to b .

Definition 4. The SALRS is anonymous if for all \mathcal{A} , $A dv_{\mathcal{A}}^{\text{ano}}$ is negligible, where $A dv_{\mathcal{A}}^{\text{ano}} = |\Pr[b = b'] - 1/2|$. We name the game for anonymity Game_{ano} .

4.6. Master-Public-Key-Unlinkability. Master-public-key-unlinkability means that with the support of a derived public key and the corresponding signatures, no one can distinguish which master public key is the one which it was derived from.

4.6.1. Setup. Same as that of Game_{uf} .

4.6.2. Probing Phase 1. Same as the probing phase of Game_{uf} .

4.6.3. Challenge. \mathcal{A} outputs two distinct indices $1 \leq i_0, i_1 \leq n(\lambda)$. A random bit $b \in \{0, 1\}$ is chosen, and $\text{DPK}^* \leftarrow \text{DerivedPublicKeyGen}(\text{MPK}_{i_b})$ is given to \mathcal{A} . Set $L_{dpk} = L_{dpk} \cup \{\text{DPK}^*\}$.

4.6.4. Probing Phase 2. Same as the probing phase 1, with the restriction that ODPKAdd $(\text{DPK}^*, \text{MPK}_{i_j})$ ($j \in \{0, 1\}$) cannot be queried.

4.6.5. Output Phase. \mathcal{A} outputs a bit $b' \in \{0, 1\}$ as its guess to b .

Definition 5. The SALRS is master-public-key-unlinkable if for all \mathcal{A} , $A dv_{\mathcal{A}}^{\text{mpkunkl}}$ is negligible, where $A dv_{\mathcal{A}}^{\text{mpkunkl}} = |\Pr[b = b'] - 1/2|$. We name the game for master-public-key-unlinkability $\text{Game}_{\text{mpkunkl}}$.

4.7. Derived-Public-Key-Unlinkability. Derived-public-key-unlinkability means that with the support of two derived public keys and the corresponding signatures, no one can figure out whether they are derived from the same master public key.

4.7.1. Setup. Same as that of Game_{uf} .

4.7.2. Probing Phase 1. Same as the probing phase of Game_{uf} .

4.7.3. Challenge. \mathcal{A} outputs two distinct indices $1 \leq i_0, i_1 \leq n(\lambda)$. A random bit $c \in \{0, 1\}$ is chosen. Compute $\text{DPK}_0^* \leftarrow \text{DerivedPublicKeyGen}(\text{MPK}_{i_c})$.

A random bit $b \in \{0, 1\}$ is chosen. If $b=0$, compute $\text{DPK}_1^* \leftarrow \text{DerivedPublicKeyGen}(\text{MPK}_{i_c})$; otherwise, compute $\text{DPK}_1^* \leftarrow \text{DerivedPublicKeyGen}(\text{MPK}_{i_{1-c}})$. $(\text{DPK}_0^*, \text{DPK}_1^*)$ are given to \mathcal{A} . Set $L_{dpk} = L_{dpk} \cup \{\text{DPK}_0^*, \text{DPK}_1^*\}$.

4.7.4. Probing Phase 2. Same as the probing phase 1, with the restriction that $\text{ODPKAdd}(\text{DPK}_j^*, \text{MPK}_{i_k})$ ($j, k \in \{0, 1\}$) can only be queried on at most one $j \in \{0, 1\}$.

4.7.5. Output Phase. \mathcal{A} outputs a bit $b' \in \{0, 1\}$ as its guess to b .

Definition 6. The SALRS is derived-public-key-unlinkable, if for all \mathcal{A} , $A d_{\mathcal{V}_{\mathcal{A}}^{\text{dpkunl}}}$ is negligible, where $A d_{\mathcal{V}_{\mathcal{A}}^{\text{dpkunl}}} = |\Pr[b = b'] - 1/2|$. We name the game for derived-public-key-unlinkability $\text{Game}_{\text{dpkunl}}$.

5. Our Concrete Scheme of SALRS

In this section, as a building block for our SALRS construction, we first introduce our novel concrete key encapsulation mechanism (KEM) based on the elliptic curve discrete logarithm. Then, we propose our concrete SALRS construction.

5.1. KEM Based on Elliptic Curve Discrete Logarithm. Formally, our novel concrete KEM based on the elliptic curve discrete logarithm consists of algorithms as follows.

5.1.1. Setup (1^λ) \rightarrow Params. The input to this algorithm is a security parameter 1^λ and outputs system global parameters params.

The params are generated as follows. Let $\mathbb{G}_1, \mathbb{G}_2$, and \mathbb{G}_T be groups of prime order p , \mathbb{Z}_p be an integer group of order p , g_i be a generator of ($i=1, 2$), and $e: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a bilinear, efficient, and computable map. Set $\mathbb{G}_1 \neq \mathbb{G}_2$, and there exists no efficient homomorphism between \mathbb{G}_1 and \mathbb{G}_2 . $H: \mathbb{G}_T \rightarrow \mathbb{Z}_p$ is a collision-resistant hash function. Then, we set $\text{params}=(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbb{Z}_p, g_1, g_2, H)$.

5.1.2. KeyGen (params) \rightarrow (pk, sk). The input to this algorithm is the params and output a (public key, secret key) pair (pk, sk).

The pair (pk, sk) is generated as follows. First, choose a random $\alpha \in \mathbb{Z}_p$ and then compute g_1^α . Finally, the pair (pk, sk) is set as $(\text{pk}, \text{sk})=(g_1^\alpha, \alpha)$.

5.1.3. Encaps (pk, params) \rightarrow (AD, K). The input to this algorithm is the pk and params, and output a ciphertext AD and a key K. We let AD and \mathcal{K} denote the ciphertext space and key space, respectively.

The pair (AD, K) is generated as follows. First, choose a random $r \in \mathbb{Z}_p$, compute the key g_1^r , compute $HV \triangleq H(e(\text{pk}, g_2^r))$, and then compute g_1^{HV} . Finally, set $(AD, K)=(g_1^{HV}, g_1^r)$.

5.1.4. Decaps (params, AD, pk, sk) \rightarrow K/ \perp . The input to this algorithm is the params, ciphertext AD, public key pk, and secret key sk and outputs a key K or a special symbol \perp to indicate rejection.

The K/ \perp is generated as follows. If $\exists r \in \mathbb{Z}_p$, the equation $AD = g_1^{SHV}$ holds, where $SHV \triangleq H(e(g_1^r, g_2^{sk}))$, output a key $K = g_1^r$; otherwise, output a special symbol \perp .

5.2. Concrete SALRS Construction

5.2.1. Setup (λ) \rightarrow PP. The input to this algorithm is a security parameter λ , the algorithm sets the parameters $n, q, k, l, m, \eta, \gamma$, and θ , let $H_A: \{0, 1\}^* \mapsto \mathbb{R}R_q^{k \times l}$, $\text{expandV}: \mathcal{X} \mapsto S_\eta^l, H_\theta: \{0, 1\}^* \mapsto \mathbf{B}_\theta$, and $H_m: R_q^k \mapsto R_q^{m \times l}$ be functions which are random oracles. The algorithm runs:

- (1) Set $\mathbf{A} \triangleq H_A(\text{cstr})$, where cstr is a random string belonging to $\{0, 1\}^*$
- (2) Run $\text{params} \leftarrow \text{KEM}\text{-setup}(1^\lambda)$.

Output the public parameters, $\text{PP} = (n, q, k, l, m, \eta, \gamma, \theta, H_A, \text{cstr}, \mathbf{A}, \text{KEM}, \text{params}, \text{expandV}, H_\theta, H_m)$. PP are implicit input parameters to every algorithm as follows.

5.2.2. MasterKeyGen (PP) \rightarrow (MPK, MSK). The input to this algorithm is the PP; the algorithm runs:

- (1) $(\text{pk}, \text{sk}) \leftarrow \text{KEM}\text{-KeyGen}(\text{params})$
 - (2) Set $\mathbf{t} \leftarrow \mathbf{As}$, where $\mathbf{s} \xleftarrow{R} S_\eta^l$
- Output $\text{MPK} \triangleq (\text{pk}, \mathbf{t})$ and $\text{MSK} \triangleq (\text{sk}, \mathbf{s})$.

5.2.3. DerivedPublicKeyGen (MPK) \rightarrow DPK. The input to this algorithm is the $\text{MPK} = (\text{pk}, \mathbf{t})$; the algorithm runs:

- (1) Run $(AD, K) \leftarrow \text{KEM}\text{-Encaps}(\text{pk}, \text{params})$.
 - (2) Set $\mathbf{s}' \triangleq \text{expandV}(K) \in S_\eta^l, \mathbf{t}' \leftarrow \mathbf{As}'$, and $\hat{\mathbf{t}} \leftarrow \mathbf{t} + \mathbf{t}'$.
- Output $\text{DPK} \triangleq (AD, \hat{\mathbf{t}})$.

5.2.4. DerivedPublicKeyOwnerCheck (DPK, MPK, MSK) \rightarrow 1/0. The input to this algorithm is a DPK, and pair

(MPK, MSK) with $\text{MPK} = (pk, \mathbf{t})$ and $\text{MSK} = (sk, \mathbf{s})$; the algorithm runs:

- (1) If $\text{DPK} \in \text{AD} \times R_q^k$, set $\text{DPK} \triangleq (AD, \hat{\mathbf{t}}) \in \text{AD} \times R_q^k$; otherwise, return 0.
- (2) Run $K \leftarrow \text{KEM} \cdot \text{Decaps}$ (params, AD, pk, sk).
- (3) Set $\mathbf{s}' \triangleq \text{expandV}(K)$ and $\mathbf{t}' \leftarrow \mathbf{A}\mathbf{s}'$.

If $\hat{\mathbf{t}} = \mathbf{t} + \mathbf{t}'$, return 1; otherwise, return 0.

5.2.5. *DerivedPublicKeyPublicCheck (DPK) \rightarrow 1/0.* The input to this algorithm is DPK; the algorithm runs: if $\text{DPK} \in \text{AD} \times R_q^k$, return 1; otherwise, return 0.

5.2.6. *Sign (M, R, DPK, MPK, MSK) \rightarrow σ .* The input to this algorithm is a message M , a ring of well-formed derived public keys $R = (\text{DPK}_1, \dots, \text{DPK}_r)$, a derived public key $\text{DPK} \in R$, and the master key pair for DPK, where $\text{MPK} = (pk, \mathbf{t})$ and $\text{MSK} = (sk, \mathbf{s})$; the algorithm runs:

- (1) Set $\text{DPK}_i \triangleq (AD_i, \hat{\mathbf{t}}_i) \in \text{AD} \times R_q^k$ and $\mathbf{H}_i \triangleq H_m(\hat{\mathbf{t}}_i)$ ($i = 1, \dots, r$).
- (2) Let \bar{i} be $\text{DPK} = \text{DPK}_{\bar{i}} = (AD_{\bar{i}}, \hat{\mathbf{t}}_{\bar{i}})$, run $K \leftarrow \text{KEM} \cdot \text{Decaps}$ (params, $AD_{\bar{i}}$, pk, sk). Set $\mathbf{s}'_{\bar{i}} \triangleq \text{expand}(K)$ and $\hat{\mathbf{s}}_{\bar{i}} \leftarrow \mathbf{s} + \mathbf{s}'_{\bar{i}}$.
- (3) Use $\mathbf{H}_{\bar{i}}$ and $\hat{\mathbf{s}}_{\bar{i}}$ above, and set $\mathbf{I} \leftarrow \mathbf{H}_{\bar{i}} \hat{\mathbf{s}}_{\bar{i}}$.
- (4) Set $\mathbf{w}_i \leftarrow \mathbf{A}\mathbf{y}$ and $\mathbf{v}_i \leftarrow \mathbf{H}_i \mathbf{y}$, where $\mathbf{y} \leftarrow S_{\gamma}^l$.
- (5) Set $c_i \leftarrow H_{\theta}(M, R, \mathbf{w}_{i-1}, \mathbf{v}_{i-1}, \mathbf{I})$, where we set $c_1 \leftarrow H_{\theta}(M, R, \mathbf{w}_r, \mathbf{v}_r, \mathbf{I})$, and set $\mathbf{w}_i \leftarrow \mathbf{A}\mathbf{z}_i$, $c_i \hat{\mathbf{t}}_i$ and $\mathbf{v}_i \leftarrow \mathbf{H}_i \mathbf{z}_i - c_i \mathbf{I}$, where $\mathbf{z}_i \leftarrow S_{\gamma-2\theta\eta}^l$, $i = \bar{i} + 1, \dots, r, 1, \dots, \bar{i} - 1$.
- (6) Set $c_{\bar{i}} \leftarrow H_{\theta}(M, R, \mathbf{w}_{\bar{i}-1}, \mathbf{v}_{\bar{i}-1}, \mathbf{I})$.

Set $\langle b \rangle \mathbf{z}_i \leftarrow \mathbf{y} \langle b \rangle + c_i \hat{\mathbf{s}}_{\bar{i}}$. If $\mathbf{z}_i \in S_{\gamma-2\theta\eta}^l$, output $\sigma \triangleq (cc_1, \{\mathbf{z}_i\}_{i=1}^r, \mathbf{I}) \in \mathbf{B}_{\theta} \times (S_{\gamma-2\theta\eta}^l)^r \times R_q^m$; otherwise, return to (4).

5.2.7. *Verify (M, R, σ) \rightarrow 1/0.* The input to this algorithm is a message M , a ring of well-formed derived public keys $R = (\text{DPK}_1, \dots, \text{DPK}_r)$, and a signature $\sigma = (c_1, \{\mathbf{z}_i\}_{i=1}^r, \mathbf{I})$; the algorithm runs:

- (1) If $c_1 \notin \mathbf{B}_{\theta}$ or $\mathbf{z}_i \notin S_{\gamma-2\theta\eta}^l$, $\exists i \in \{1, \dots, r\}$, return 0.
- (2) Set $\text{DPK}_i \triangleq (AD_i, \hat{\mathbf{t}}_i) \in \text{AD} \times R_q^k$ and $\mathbf{H}_i \triangleq H_m(\hat{\mathbf{t}}_i)$ ($i = 1, \dots, r$). Then, set $\langle b \rangle \mathbf{w}_i \leftarrow \mathbf{A}\mathbf{z}_i - \langle b \rangle c_i \hat{\mathbf{t}}_i$, $\langle b \rangle \mathbf{v}_i \leftarrow \mathbf{H}_i \langle b \rangle \mathbf{z}_i - c_i \mathbf{I}$, and $c_{i+1} \leftarrow H_{\theta}(M, R, \mathbf{w}_i, \mathbf{v}_i, \mathbf{I})$.

If $c_{r+1} = c_1$, return 1; otherwise, return 0.

5.2.8. *Link ($M_0, R_0, \sigma_0, M_1, R_1, \sigma_1$) \rightarrow 1/0.* The input to this algorithm is two valid (message M , derived public key ring R , and signature σ) tuples (M_0, R_0, σ_0) and (M_1, R_1, σ_1), where $\sigma_0 = (c_1^{(0)}, \{\mathbf{z}_i^{(0)}\}_{i=1}^{r_0}, \mathbf{I}^{(0)})$ and $\sigma_1 = (c_1^{(1)}, \{\mathbf{z}_i^{(1)}\}_{i=1}^{r_1}, \mathbf{I}^{(1)})$; the algorithm runs: if $\mathbf{I}^{(0)} = \mathbf{I}^{(1)}$, return 1; otherwise, return 0.

6. Security Analysis of Our SALRS Construction

Now, we prove that our construction has the usual properties for a SALRS such as correctness, unforgeability, anonymity, linkability, nonslanderability, master-public-key-unlinkability, and derived-public-key-unlinkability.

6.1. *Correctness Analysis.* It is obvious that from our SALRS construction, (1) and (2) of correctness are satisfied. Therefore, we next prove (3) of correctness. Let $\sigma_j = (c_1^{(j)}, \{\mathbf{z}_i^{(j)}\}_{i=1}^{r_j}, \mathbf{I}^{(j)})$ be generated by $\text{sign}(M_j, R_j, \text{DPK}_j, \text{MPK}_j, \text{MSK}_j)$ ($j = 0, 1$) and let $\text{DPK}_j = (AD_j, \hat{\mathbf{t}}_j)$.

- (1) If $\text{DPK}_0 = \text{DPK}_1$, we have $\hat{\mathbf{s}}_0 = \hat{\mathbf{s}}_1$, and then $\mathbf{I}^{(0)} = \mathbf{I}^{(1)}$. In this case, we have link outputs 1.
- (2) If $\text{DPK}_0 \neq \text{DPK}_1$, we now prove that link outputs 0 with overwhelming probability. If $\hat{\mathbf{t}}_0 \neq \hat{\mathbf{t}}_1$, we can see that $\mathbf{H}_m(\hat{\mathbf{t}}_0), \mathbf{H}_m(\hat{\mathbf{t}}_1)$ are distinct. $\hat{\mathbf{s}}_0$ and $\hat{\mathbf{s}}_1$ are distinct. Then we have the result that the probability of $\mathbf{I}^{(0)} = \mathbf{H}_m(\hat{\mathbf{t}}_0) \hat{\mathbf{s}}_0 =$

$\mathbf{H}_m(\hat{\mathbf{t}}_1) \hat{\mathbf{s}}_1 = \mathbf{I}^{(1)}$ is negligible. If $\hat{\mathbf{t}}_0 = \hat{\mathbf{t}}_1$ but $AD_0 \neq AD_1$, we want to prove $\hat{\mathbf{s}}_0 = \hat{\mathbf{s}}_1$. We consider $\hat{\mathbf{s}}_0 \neq \hat{\mathbf{s}}_1$. If $\hat{\mathbf{s}}_0 \neq \hat{\mathbf{s}}_1$ and $\hat{\mathbf{t}}_0 = \mathbf{A} \hat{\mathbf{s}}_0 = \mathbf{A} \hat{\mathbf{s}}_1 = \hat{\mathbf{t}}_1$, its probability is negligible, so we must have $\hat{\mathbf{s}}_0 = \hat{\mathbf{s}}_1$. $\hat{\mathbf{s}}_0 = \hat{\mathbf{s}}_1$ have two cases.

- (1) $\mathbf{s}_0 \neq \mathbf{s}_1$ and $\mathbf{s}'_0 \neq \mathbf{s}'_1$:

The probability of this scenario is negligible because of the randomness of $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}'_0$, and \mathbf{s}'_1 .

- (2) $\mathbf{s}_0 = \mathbf{s}_1$ and $\mathbf{s}'_0 = \mathbf{s}'_1$:

The probability of $\mathbf{s}_0 = \mathbf{s}_1$ is negligible because two different executions of algorithm $\text{DerivedPublicKeyGen}$ with $AD_0 \neq AD_1$ will produce distinct $\mathbf{s}'_0 = \mathbf{s}'_1$ with overwhelming probability. This completes the correctness analysis.

6.2. *Security Analysis.* We use \mathcal{A} to denote any probabilistic polynomial time (PPT) adversary in security games.

Theorem 1. *The SALRS construction is linkable.*

Proof. We now prove that our SALRS construction is linkable under module-SIS assumption. If \mathcal{A} succeeds because (3) of linkability holds, it means that $\exists i, j \in [1, k]$ and $i \neq j$, $\text{DPK}_i = \text{DPK}_j$ where $\text{DPK}_i \in R_i^*$ and $\text{DPK}_j \in R_j^*$. Then, we set $\text{DPK}_i = (AD_i, \hat{\mathbf{t}}_i)$ and $\text{DPK}_j = (AD_j, \hat{\mathbf{t}}_j)$, and we have $\hat{\mathbf{t}}_i = \hat{\mathbf{t}}_j = \mathbf{A} \hat{\mathbf{s}}_j = \mathbf{A} \hat{\mathbf{s}}_i$. With the support of module-SIS assumption, we have $\hat{\mathbf{s}}_i = \hat{\mathbf{s}}_j$ with overwhelming probability, which also means $\text{DPK}_i = \text{DPK}_j$ with overwhelming probability. From (1) of linkability and (1) of correctness, we have $\sigma_i^* = \text{sign}(M_i^*, R_i^*, \text{DPK}_i, \text{MPK}_i, \text{MSK}_i)$ and $\sigma_j^* = \text{sign}(M_j^*, R_j^*, \text{DPK}_j, \text{MPK}_j, \text{MSK}_j)$. Finally, from (3) of correctness, we can find that (2) of linkability is not satisfied. This completes the proof. \square

Theorem 2. *The SALRS construction is nonslanderable.*

Proof. We now prove that our SALRS construction is nonslanderable under the correctness of the SALRS scheme. If \mathcal{A} succeeds, from (1) and (2) of nonslanderability and (1) of correctness, we have $\sigma^* = \text{sign}(M^*, R^*, \text{DPK}^*, \text{MPK}^*, \text{MSK}^*)$ and $\sigma' = \text{sign}(M', R', \text{DPK}', \text{MPK}', \text{MSK}')$. Because of (4) of nonslanderability and (3) of correctness, we have $\text{DPK}^* = \text{DPK}'$ with overwhelming probability. So, we can find that (3) of nonslanderability is not satisfied. This completes the proof. \square

Lemma 1. (See [9]). *If a SALRS scheme is linkable and nonslanderable, then it is unforgeable.*

Theorem 3. *The SALRS construction is unforgeable.*

Proof. According to Theorem 1, Theorem 2, and Lemma 1, our SALRS scheme is unforgeable. This completes the proof. \square

Theorem 4. *The SALRS construction is anonymous.*

Proof. We now prove that our SALRS construction is anonymous under the decision module-LWE assumption. If \mathcal{A} succeeds, we set $\sigma = (c_1, \{z_i\}_{i=1}^r, \mathbf{I})$ and $\text{DPK}_{i_b} = (AD_{i_b}, \hat{\mathbf{t}}_{i_b})$ ($b \in \{0, 1\}$). From algorithm sign , we have $\mathbf{I} = \mathbf{H}_{i_b} \hat{\mathbf{s}}_{i_b}$. It means that \mathcal{A} can distinguish $\mathbf{H}_{i_b} \hat{\mathbf{s}}_{i_b}$ and $\mathbf{H}_{i_b} \hat{\mathbf{s}}_{i_b}$, which contradicts the decision module-LWE assumption. This completes the proof. \square

Theorem 5. *The SALRS construction is master-public-key-unlinkable.*

Proof. We now prove that our SALRS construction is master-public-key-unlinkable under the BDH-1 assumption. If \mathcal{A} succeeds, we set $\text{DPK}^* = (AD, \hat{\mathbf{t}})$, $AD = AD_i$, and $\hat{\mathbf{t}} = \hat{\mathbf{t}}_i$ ($i \in \{0, 1\}$), that means that \mathcal{A} can distinguish $(AD_i, \hat{\mathbf{t}}_i)$ with a nonnegligible probability. From the algorithm $\text{DerivedPublicKeyGen}$, we have $\hat{\mathbf{t}}_i = \mathbf{t}_i + \mathbf{t}'_i$, $\mathbf{t}'_i = \mathbf{A}\mathbf{s}'_i$, $\mathbf{s}'_i = \text{expandV}(K)$, and $(AD, K) \leftarrow \text{KEM} \cdot \text{Encaps}(\text{pk}, \text{params})$, where $K = K_i$. It is obvious that because of the randomness of $r \in \mathbb{G}_1$ and $K = g_1^r$ in the algorithm Decaps , \mathcal{A} cannot distinguish K with an overwhelming probability. Therefore, \mathcal{A} cannot distinguish \mathbf{s}'_i , so \mathcal{A} cannot distinguish $\hat{\mathbf{t}}_i$ with an overwhelm probability.

We now prove that \mathcal{A} cannot distinguish AD_i with an overwhelming probability too. If \mathcal{A} can distinguish AD_i with a nonnegligible probability ϵ , we can construct a PPT algorithm \mathcal{B} that solves the BDH-1 problem. To be specific, we assume that \mathcal{A} and \mathcal{B} play the game Game_{sec} , and \mathcal{B} simulates the challenger and tries to solve the BDH-1 problem. Suppose the BDH-1 instance (g_1, g_2, g_1^a, g_1^b) is given to \mathcal{B} . \mathcal{B} initializes system parameters and gets $\text{params} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbb{Z}_p, g_1, g_2, H)$ from KEM. Then, \mathcal{B} interacts with \mathcal{A} as follows. \square

6.2.1. *Setup.* \mathcal{B} sends params to \mathcal{A} .

6.2.2. *Query 1*

- (1) Key pair query: \mathcal{A} asks \mathcal{B} to use the algorithm KeyGen of KEM to compute a key pair $(\text{pk}, \text{sk}) = (g_1^\alpha, \alpha)$ and return it to \mathcal{A} . \mathcal{A} can only query at most q_1 times for key pairs.
- (2) Public key query: \mathcal{A} ask \mathcal{B} to use the algorithm KeyGen of KEM to compute a key pair (pk, sk) and return the public key pk to \mathcal{A} . \mathcal{A} can only query at most q_2 times for key pairs.
- (3) Hash query: \mathcal{B} sets a hash list $Hlist$. $Hlist$ is initialized as an empty set. When \mathcal{A} submits a random element $g_T \in \mathbb{G}_T$ to \mathcal{B} , \mathcal{B} answers as follows. If it has not appeared in $Hlist$, \mathcal{B} choose a random element $z \in \mathbb{Z}_p$ and returns it to \mathcal{A} . Then, \mathcal{B} stores the tuple (g_T, z) in $Hlist$. Otherwise, \mathcal{B} finds out the tuple (g_T, z) and returns z to \mathcal{A} . \mathcal{A} can only query at most q_3 times for hash queries.

6.2.3. *Challenge.* \mathcal{A} chooses a public key pk^* from (1) of query 1 and sends it to \mathcal{B} . \mathcal{B} chooses a random bit $\delta \in \{0, 1\}$; if $\delta = 0$, \mathcal{B} sets $K^* = g_1^b$ and computes $AD^* \leftarrow \text{Encaps}(pk^*, \text{params})$ and then returns them to \mathcal{A} . Otherwise, \mathcal{B} chooses a random element $AD^* \in \mathbb{G}_1$ and sets $K^* = g_1^b$ and returns them to \mathcal{A} .

6.2.4. *Query 2.* \mathcal{A} can make queries as he does in query 1 except secret keys for pk^* .

6.2.5. *Guess.* Finally, \mathcal{A} outputs a bit δ' as the guess of δ .

If \mathcal{A} can distinguish AD_i , then \mathcal{A} can guess the answer. Then, \mathcal{B} chooses tuple (g_T, z) in $Hlist$, which satisfies $g_T = e(g_1, g_2)^{ab}$. Then, \mathcal{B} outputs z as the solution to BDH-1 problem. The probability that \mathcal{B} solves the BDH-1 problem is that $\Pr[\mathcal{B} \text{ succeeds}] = \epsilon \cdot \Pr[\mathcal{A}(pk^* = g_1^a)] \cdot \Pr[\mathcal{B}(z = e(g_1^a, g_1^b))]$.

If \mathcal{B} succeeds in obtaining a solution of BDH-1 problem, the following conditions must be satisfied:

- (1) $\Pr[\mathcal{A}(pk^* = g_1^a)] \geq 1/q_2$ (\mathcal{A} correctly chooses pk^*).
- (2) $\Pr[\mathcal{B}(z = e(g_1^a, g_1^b))] \geq 1/q_3$ (\mathcal{B} correctly chooses z).

Therefore, we have $\Pr[\mathcal{B} \text{ succeeds}] \geq \epsilon / (q_2 q_3)$. It means that the probability of the fact that \mathcal{B} solves BDH-1 problem is nonnegligible, which contradicts BDH-1 assumption. This completes the proof.

Lemma 2. (See [9]). *If a SALRS scheme is master-public-key-unlinkable, then it is derived-public-key-unlinkable.*

Theorem 6. *The SALRS construction is derived-public-key-unlinkable.*

Proof. According to Theorem 5 and Lemma 2, our SALRS scheme is derived-public-key-unlinkable. This completes the proof. \square

7. Efficiency Analysis

In this section, we make a comparison with the efficiency of the SALRS scheme in [9]. The parameters $n, l, k, q, m, \theta, \eta,$ and γ are set to be same as which in [9], i.e., $n = 256, l = 5, k = 3, q \approx 2^{35}$ and $q = 17 \bmod 32, m = 1, \theta = 60, \eta = 3,$ and $\gamma = 699453$. Additionally, the functions $H_A, H_m, H_\theta,$ and expandV are set to be same as which in [9], that is to say, we use SHAKE-256 to implement the functions $H_A, H_m,$ and expandV and use the algorithm `SampleInBall` to implement H_θ . Moreover, with the parameter selection above, we have the fact that in order to obtain the signature in our SALRS scheme, the signer has to run Step 4–Step 6 of algorithm `sign` at most twice. Because, the probability of restarting of Step 4–Step 6, which can be easily worked out, is $(2\theta\eta/\gamma + 0.5) \approx 1 - e^{-(2n\theta\eta/\gamma)}$.

From [10, 18, 27], we can obtain an instantiation of KEM where the system global parameters `params` of KEM are set to be the public parameters of the stealth address scheme in [27]. Especially, the group \mathbb{G}_1 in our novel concrete KEM is instantiated to be the special elliptic curve called Ed25519 in [18].

The Ed25519 curve in [18] obviously tells us that in our SALRS scheme, the size of public key pk , secret key sk , or ciphertext AD is $(256-1) \times 2 = 510$ bits, $256 \times 2 = 512$ bits, or $(256-1) \times 2 = 510$ bits, respectively. On the other hand, the efficiency analysis of the SALRS scheme in [9] also tells us that its size of public key, secret key, or ciphertext is 1088 bytes, 2400 bytes, or $(1184-32) = 1152$ bytes, respectively. With the datum above, we can find that the size of public key in our SALRS scheme is smaller than that in [9], which means that from the construction of master public key (MPK), the size of MPK in our SALRS scheme is also smaller than that in [9]. The same applies to the master secret key (MSK) and derived public key (DPK). It comes out a conclusion that with regard to the size of MPK, MSK, and DPK, our SALRS scheme has less storage cost.

8. Conclusion

In this study, the linkable ring signature scheme with stealth addresses were addressed. Then, we proved the security of proposed schemes under the assumptions of BDH-1 problem, module-SIS problem, and module-LWE problem. The results showed that our schemes have all the properties that a linkable ring signature scheme with stealth addresses should have, i.e., unforgeability, anonymity, linkability, nonlanderability, master-public-key-unlinkability, and derived-public-key-unlinkability. Efficiency analysis showed that our SALRS scheme has less storage cost than the SALRS scheme in [9] under the same security conditions.

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 research was supported in part by the National Key Research and Development Program of China (2021YFA1000600), the City School Joint Funding Project of Guangzhou City (202102010377), and the National Natural Science Foundation of China (61702124).

References

- [1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008, <https://bitcoin.org/bitcoin.pdf>.
- [2] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *Ieee Access*, vol. 4, pp. 2292–2303, 2016.
- [3] T. Li, Y. Chen, Y. Wang et al., "Rational protocols and attacks in blockchain system," *Security and Communication Networks*, vol. 2020, Article ID 8839047, 11 pages, 2020.
- [4] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
- [5] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, 2021.
- [6] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "Psspr: a source location privacy protection scheme based on sector phantom routing in wsns," *International Journal of Intelligent Systems*, 2021.
- [7] S. Noether, "Ring signature confidential transactions for monero," *IACR Cryptol. ePrint Arch.* vol. 1098, p. 2015, 2015.
- [8] S.-F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen, Ringct 2.0: a compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero," in *Proceedings of the Computer Security - ESORICS 2017. In European Symposium on Research in Computer Security*, pp. 456–474, Springer, Oslo, Norway, September 2017.
- [9] Z. Liu, K. Nguyen, G. Yang, H. Wang, and D. S. Wong, "A lattice-based linkable ring signature supporting stealth addresses," in *Lecture Notes in Computer Science*, pp. 726–746, Springer, Berlin, Germany, September 2019.
- [10] N. Van Saberhagen, *Cryptonote, v. 2.0*, 2013, <https://goo.gl/kfojVZ>.
- [11] J. K. Liu, V. K. Wei, and D. S. Wong, "Linkable spontaneous anonymous group signature for ad hoc groups," in *Proceedings of the Information Security and Privacy. In Australasian Conference on Information Security and Privacy*, pp. 325–335, Springer, Sydney, Australia, 2004.
- [12] P. Todd, "Stealth addresses," *Post on Bitcoin development mailing list*, 2014, <https://www.mail-archive.com/bitcoin-development@lists.sourceforge.net/msg03613.html>.
- [13] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou, "Linkable ring signature with unconditional anonymity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 157–165, 2013.
- [14] M. H. Au, S. S. M. Chow, W. Susilo, and P. P. Tsang, "Short linkable ring signatures revisited," in *Proceedings of the Public Key Infrastructure. In European Public Key Infrastructure Workshop*, pp. 101–115, Springer, Turin, Italy, June 2006.

- [15] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen, "Certificate based (linkable) ring signature," in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 79–92, Springer, Hong Kong, China, 2007.
- [16] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen, "Secure id-based linkable and revocable-iff-linked ring signature with constant-size construction," *Theoretical Computer Science*, vol. 469, pp. 1–14, 2013.
- [17] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky, "Lattice Signatures and Bimodal Gaussians," in *Proceedings of the Annual Cryptology Conference Advances in Cryptology - CRYPTO 2013*, pp. 40–56, Springer, Santa Barbara, CA, USA, August 2013.
- [18] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, "High-speed high-security signatures," *Journal of cryptographic engineering*, vol. 2, no. 2, pp. 77–89, 2012.
- [19] W. A. A. Torres, R. steinfeld, A. sakzad et al., "Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice ringct v1. 0)," in *Proceedings of the Australasian Conference on Information Security and Privacy*, pp. 558–576, Springer, Wollongong, Australia, July 2018.
- [20] C. Baum, H. Lin, and S. Oechsner, "Towards practical lattice-based one-time linkable ring signatures," in *Proceedings of the International Conference on Information and Communications Security*, pp. 303–322, Springer, Lille, France, October, 2018.
- [21] Y. Ren, H. Guan, and Q. Zhao, "An efficient lattice-based linkable ring signature scheme with scalability to multiple layer," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2021, pp. 1–10, 2021.
- [22] N. Courtois and R. Mercer, "Stealth address and key management techniques in blockchain systems," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy ICISSP*, pp. 559–566, Porto, Portugal, January 2017.
- [23] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proceedings of the Advances in Cryptology - CRYPTO 2001*, pp. 213–229, Springer, California, CA, USA, August 2001.
- [24] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *Designs, Codes and Cryptography*, vol. 75, no. 3, pp. 565–599, 2015.
- [25] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers," *Discrete Applied Mathematics*, vol. 156, no. 16, pp. 3113–3121, 2008.
- [26] M. S. Kiraz and O. Uzunkol, "Still wrong use of pairings in cryptography," 2016, https://www.researchgate.net/publication/301855144_Still_Wrong_Use_of_Pairings_in_Cryptography.
- [27] J. Fan, Z. Wang, Y. Luo, J. Bai, Y. Li, and Y. Hao, "A new stealth address scheme for blockchain," in *Proceedings of the ACM Turing Celebration Conference-China*, Chendu, China, pp. 1–7, 2019.

Research Article

Blockchain-Based Proof of Retrievability Scheme

Yan Ren ¹, Haipeng Guan ¹, Qiuxia Zhao ¹, and Zongxiang Yi ^{2,3,4}

¹School of Mathematics and Information Technology, Yuncheng University, Yuncheng 044000, China

²School of Mathematics and Systems Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China

³Guangdong Provincial Key Laboratory of Information Security Technology, Guangzhou 510006, China

⁴School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China

Correspondence should be addressed to Zongxiang Yi; tpu01yzx@gmail.com

Received 17 October 2021; Revised 15 November 2021; Accepted 24 December 2021; Published 3 February 2022

Academic Editor: Yuling Chen

Copyright © 2022 Yan Ren 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, user information is usually collected by all kinds of smart devices. The collected user information is stored in the cloud storage, and there is a risk of information leakage. In order to protect the security and the privacy of user information, the user and cloud provider will periodically execute a protocol called proof of retrievability scheme. A proof of retrievability scheme ensures the security of the data by generating proof to convince the user that the cloud provider does correctly store the user information. In this paper, we construct a proof of retrievability scheme using the blockchain technology. Using the advantage that the stored data cannot be tampered with in blockchain, this ensures the integrity of the data. Specifically, some related definitions, security models, and a blockchain-based construction of a proof of retrievability scheme are given. Then the validity and security of the scheme are proved later. As a result, user information can be protected by our scheme.

1. Introduction

1.1. Background. With the information systems coming into our life, there are many user private information appliances such as surveillance cameras, smartwatches, smart door locks, and the online supermarket. They provide a lot of convenience for our life. However, these providers will collect user information and store it in the cloud where new technologies are widely used [1–7]. Due to the vulnerability of the cloud, user information could be attacked by hackers in information systems and can be easily stolen if the cloud storage provider is compromised. Among the problems and challenges of cloud storage [8–10], only the problem of how to ensure the security and integrity of the information is considered in the paper. In order to solve it, three kinds of methods are used [11]: proof of ownership (PoW), provable data possession (PDP), and proof of retrievability (PoR). We focus on the PoR and for the state-of-the-art of PoR, the reader is referred to [11–25].

Generally, the schemes of PoR are under different settings and security models. On the one hand, some schemes [11–14] are for static data. Some schemes [15–17] discussed

the multiserver setting. In these schemes, the client can identify machines and recover the data from the others by using the audit mechanism. Other schemes [18–25] are for dynamic data. On the other hand, works in [18–21] are about security. The authors of [22–24] researched on memory checking and study how to authenticate remotely stored dynamic data. The scheme in [25] is for the multiserver and dynamic data setting.

Recently, blockchain is used to eliminate a trusted third party in many protocols [26]. However, it is still unknown how to utilize blockchain in PoR schemes, which is also a new challenge in constructing a PoR scheme.

1.2. Motivation and Contribution. The concept of blockchain was first proposed in 2008 in “Bitcoin: a peer-to-peer electronic cash system” [27] published by the cryptography mailing group by a scholar known by the pseudonym “Satoshi Nakamoto.” The verification, bookkeeping, storage, maintenance, and transmission of the data in blockchain are all based on the distributed system structure, and the trust relationship between distributed nodes is established by the

pure mathematical method instead of the central mechanism. Thus, a decentralized and reliable distributed system can be formed. The goal of blockchain is to provide trusty for transactions between untrusted entities, without the need for a trusted third party. At present, many institutions have combined the industry conditions with the characteristics of blockchain and made beneficial attempts in many industries, including payment, Internet of things, credit investigation, transaction settlement and clearing, crowdfunding, equity transaction, audit, supply chain, digital asset management, notarization, and other fields [28–33]. We consider using blockchain technology to solve the problem of the trusted third party in the verification of the PoR scheme.

In this paper, we first define a security model for the blockchain-based proof of retrievability by modifying the model in [14, 34, 35]. Secondly, we propose the first concrete PoR scheme based on blockchain. Finally, we demonstrate that the proposed scheme is provably secure in the new model.

1.3. Organization. The rest of the paper is organized as follows. Preliminaries are given in Section 2. In Section 3, we formally define the framework and security model for blockchain-based PoR schemes. Then a concrete construction of a blockchain-based scheme is presented in Section 4. We analyze the security of the proposed scheme in Section 5. Finally, conclusions are made in Section 6.

2. Preliminaries

In this section, some notions are introduced such as hash function, Merkle tree, blockchain, and bilinear pairing.

2.1. Hash Function. The hash function H is used to map data x of an arbitrary length (input) to data $y = H(x)$ of fixed length (output). y is called the hash of x . Many Hash functions [36] are widely publicly available and can be selected based on the context.

$$H: \{0, 1\}^* \longrightarrow \{0, 1\}^n. \quad (1)$$

This transformation is a compression mapping, which has the following properties:

- (i) The space of the hash value is usually much smaller than the space of the input.
- (ii) Different inputs may hash into the same output, but it is hard to find two different inputs x, x' such that $H(x) = H(x')$.
- (iii) It is infeasible to determine the input value x from the hash value y .

Assumption 1 (hash function preimage assumption). Given $y = H(x)$, it is hard to compute x .

Assumption 2 (hash function collision assumption). Given x , it is hard to compute x' such that $H(x) = H(x')$.

2.2. Merkle Tree. Merkle tree, also known as a Hash tree, as the name implies, is a tree that stores hash values. A leaf node of a Merkle tree is attached to the hash value for a data block. A nonleaf node is attached to the cryptographic hash of its corresponding child nodes.

Figure 1 presents a simple example of a Merkle tree with 4 pieces of data. Let f be a hash function and $X = \{x_0, x_1, x_2, x_3\}$ denotes the set of data used to generate the Merkle tree. A Merkle tree is generated as follows: firstly, for all leaf nodes, $y_{\text{bin}(i)} = f(x_i)$ where $i = 1, 2, 3, 4$ and $\text{bin}(i)$ is the binary form of i ; secondly, for all inside nodes, the value of the node is $f(y_l \| y_r)$ where y_l and y_r are the value of left child and right child, respectively. An Merkle tree is **valid** if and only if the value of each inside node equals to $f(y_l \| y_r)$. As a result, this example outputs the following:

$$\begin{aligned} y_{0,0} &= f(x_0), y_{0,1} = f(x_1), y_{1,0} = f(x_2), y_{1,1} = f(x_3), \\ y_0 &= f(y_{0,0} \| y_{0,1}), y_1 = f(y_{1,0} \| y_{1,1}), \\ y &= f(y_0 \| y_1). \end{aligned} \quad (2)$$

In a Merkle tree, the value of the root node is called the hash of the Merkle tree. For the example in Figure 1, the hash of that tree with data X is

$$y = f(f(f(x_0) \| f(x_1)) \| f(f(x_2) \| f(x_3))). \quad (3)$$

In the rest of this paper, we use $\text{Merkel}(X)$ to denote the Merkle tree created by the data set X and use $H(T)$ to denote the hash of a Merkle tree T , where H is the underlying hash function. For example, the hash of the Merkle tree created by the data set X can be denoted by $H(\text{Merkel}(X))$.

2.3. Blockchain. Within a blockchain, the hash function is used to determine the state of the blockchain and Figure 2 shows the structure of blockchain which can be viewed as a linked list of blocks. Every block has four basic objects: the hash of the previous block, the timestamp of generation, the random number of security, and the hash of a Merkle tree. Usually, the corresponding Merkle tree is linked with the block too. Two neighbor blocks are linked by a hash pointer that points from the previous block and thus it creates a chain of connected blocks, hence the name blockchain. By linking blocks in this manner, the ordered hashes of all the n blocks represent the entire state of the blockchain, namely,

$$f(f(\text{Block}(0) \| f(\text{Block}(1)) \| \dots \| f(\text{Block}(n))), \quad (4)$$

where f is a hash function. A blockchain is **valid** if $f(\text{Block}(i-1))$ equal to the value of the field *hash* of $\text{Block}(i-1)$ in the structure of the block $\text{Block}(i)$, for all $1 \leq i \leq n$.

To utilize blockchain for a data set X (see the example in Subsection 2.2), a corresponding Merkle tree T will be constructed by the data set X . Then a new block B denoted by $B(X)$ can be generated with the help of a timestamp provider. Adding more parameters, we use $B(X; ts)$ to denote a block where X is the data set to generate the hash of the Merkle tree, ts is the timestamp of the current time, and

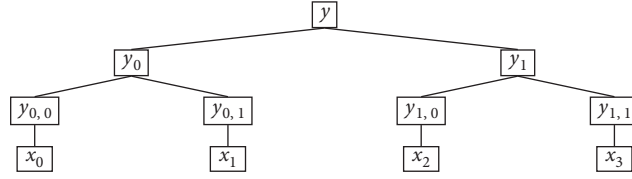


FIGURE 1: The structure of the Merkle tree.

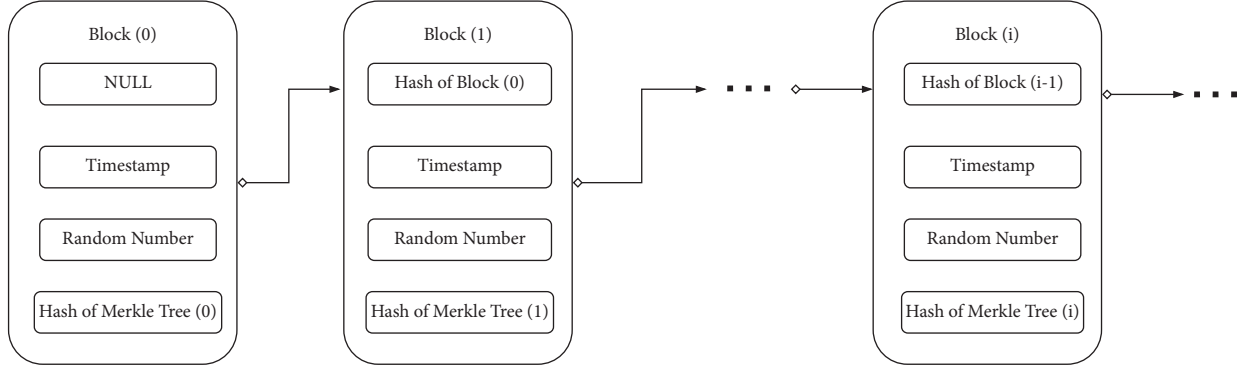


FIGURE 2: The structure of blockchain.

rand is the random number. Moreover, a blockchain provides the following operations:

- (i) **NewBlock(X;ts)**: create a valid block $B(X; ts)$.
- (ii) **AppendBlock(B)**: append the block B to the blockchain by filling a suitable random number in the block.
- (iii) **FetchBlock(ts)**: return the block at a time ts in the blockchain. If there are no blocks at that time, then *NULL* is returned.

Recently, there are issues in maintaining a blockchain, such as generating blocks [37, 38] and updating with efficiency [39]. Anyway, to summarize the characteristic of blockchain, we have the following assumption.

Assumption 3 (blockchain assumption). All the state and blocks of blockchain is hard to modify after they were generated.

2.4. Bilinear Pairing. Bilinear pairing is also called bilinear mapping, which was first used to construct tripartite key exchange protocol [40]. It involves three multiplicative cyclic groups $G_1, G_2,$ and G_T which have a prime order p . Bilinear pairing is a mapping $e: G_1 \times G_2 \rightarrow G_T$ satisfying the following conditions:

- (1) For any $g_1 \in G_1, g_2 \in G_2,$ and $a, b \in \mathbb{Z}_p,$ it always has $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$
- (2) There exists two elements $g_1 \in G_1$ and $g_2 \in G_2$ such that $e(g_1, g_2) \neq 1_{G_T}$ where 1_{G_T} is the identity in G_T
- (3) For any $g_1 \in G_1, g_2 \in G_2,$ it is feasible to compute $e(g_1, g_2)$

Let $c_1, c_2, c_3 \in \mathbb{Z}_p$ and g_1, g_2, g be the generators of $G_1, G_2, G_T,$ respectively. There are two security assumptions related to bilinear pairing.

Assumption 4 (bilinear decisional Diffie–Hellman). Given a bilinear pairing $e, g_1^{c_1} \in G_1, g_2^{c_2} \in G_2, g^{c_3} \in G_T, e(g_1, g_2)^{c_1 c_2 c_3}$ and a randomly selected element $T \in G_T,$ it is hard to distinguish $e(g_1, g_2)^{c_1 c_2 c_3}$ from T .

Assumption 5 (bilinear computational Diffie–Hellman). Given a bilinear pairing $e, g_1^{c_1} \in G_1, g_2^{c_2} \in G_2, g^{c_3} \in G_T,$ it is hard to compute $e(g_1, g_2)^{c_1 c_2 c_3}$.

3. Security Model

3.1. System Setting. Our system has three entities, the user, the cloud storage provider where user information is stored, and a blockchain where several timestamp providers are available to all entities. The structure of the system setting is shown in Figure 3.

- (i) **The User.** The user is the entity who wants to store the data on the cloud storage. Whenever the user wants to check whether the data is correctly stored on the cloud storage, then a request of PoR will be generated and sent to the cloud storage. With the help of blockchain, the user can verify the retrievability of stored data by the proof received from the cloud storage provider.
- (ii) **The Cloud Storage Provider.** A Cloud storage provider is an entity who exactly stores the data for the user. Besides, the cloud storage provider generates and sends the proof of retrievability after receiving the request from the user.

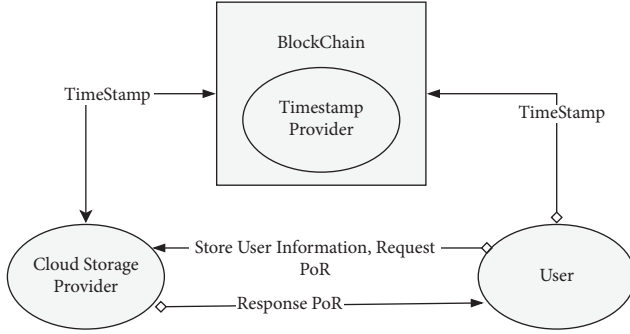


FIGURE 3: System setting.

- (iii) *BlockChain*. BlockChain is mainly for keeping the transcripts of PoR scheme constant. Moreover, timestamp providers in a blockchain can help the cloud storage providers to generate PoR and the user to verify the generated PoR.

3.2. Timestamp Usage. The timestamps are provided to both the user and the cloud storage provider. The existence of the data is guaranteed by timestamp through computing the hash value which is included in the next timestamp. In our scheme, we will modify the traditional timestamp computation. At the end of every proof generation, the timestamp provider proceeds to compute a timestamp on the current time and makes the timestamp published on the blockchain. The timestamp is used to compute the hash value in blockchain by the cloud storage provider.

We benefit the security from the usage of timestamps. On the one hand, running a PoR scheme twice at two different moments would be the PoR for the duration between the two moments. On the other hand, it gives a timeline of PoR records which can be used to analyze the efficiency.

3.3. Definition. There are five algorithms in the blockchain-based PoR which are described as follows:

- (i) *Keygen*: The input of the algorithm is the security parameter, and the output is the public key and private key of the system and the user.
- (ii) *Outsource*: In this stage, it inputs the private key and user data M , and outputs a data set Y with n blocks and one tag σ for each block. For the blockchain, also generates new blocks for the data.
- (iii) *RequestChallenge*: The user randomly selects a challenge r and sends it to the cloud storage provider.
- (iv) *ResponseProof*: The proof process is an interactive protocol. The input is a public key, the file name and tag of the file and the output is *proof* for a proof response.
- (v) *VerifyProof*: The input is a system parameter and *proof*, the output of the algorithm is *accepted* or *rejected*.

Remark 1. Note that system parameter includes the structure and the state of selected blockchain, as well as another luxury public information such as the hash function implementations and bilinear pairing implementations.

3.4. Security Model. Under the assumptions mentioned in Section 2, a blockchain-based PoR scheme is secure if it satisfies the following two properties.

- (1) *Correctness*. If all the effective proofs generated by the algorithm (*KeyGen*, *outsourcing*, *Request Challenge*, *Response Proof*, and *Verify Proof*) are defined above, the verification algorithm outputs *accept*, then a blockchain-based PoR scheme is correct.
- (2) *Reasonableness*. For reasonableness, if any malicious cloud storage provider can generate proof such that the *Verify Proof* outputs *accept*. That is, the user believes that the cloud storage provider can generate the proof only if it correctly stores the user data.

If the probability that an adversary with arbitrary probabilistic polynomial-time wins the game described below is negligible, then a blockchain-based PoR scheme is reasonableness.

- (a) *Setup*: The challenger runs the *Keygen* algorithm to obtain the public key and private. Then the public key is sent to the adversary.
- (b) *Outsource*: The adversary selects a data set and sends it to the challenger, who runs the *Outsourcing* algorithm and responds with the output.
- (c) *ChallengeProof*:
 - (1) In the *Request Challenge* algorithm, the challenger randomly generates a challenge message and sends it to the adversary.
 - (2) The adversary generates a data set first by running an arbitrary algorithm that returns a proof. The proof will be sent to the challenger in the *Response Proof* algorithm.
- (d) *Verify*: The challenger runs the *VerifyProof* algorithm to verify the proof received from the adversary. It outputs *accept* if and only if the proof is accepted by the challenger.

The adversary wins the game if *accept* is outputted in the last *Verify* step.

4. Our PoR Scheme

4.1. High Description. In this section, we will propose a blockchain-based PoR scheme. To cut costs, the cloud storage provider only needs to generate a Merkle tree for a data set and store the hash of the Merkle tree in the blockchain. The data set can be stored anywhere by the cloud storage provider. When the user requests a challenge of PoR, the cloud storage provider fetches back the Merkle tree and generates a PoR to the user with the help of blockchain.

4.2. *Blockchain-Based Proof of Retrievability Scheme.* Our scheme consists of five algorithms, namely, *Keygen*, *Outsource*, *RequestChallenge*, *ResponseProof*, and *VerifyProof*.

4.2.1. *Keygen.* Both the user and the cloud storage provider make consensus on public system parameters: a hash function H , a blockchain BC, a block size t , a prime number p , a generator g of the cyclic multiplicative group (\mathbb{Z}_p, \times) , and a bilinear pairing e on \mathbb{Z}_p .

The user chooses a nonzero element $s \in \mathbb{Z}_p$ randomly as a private key and computes and public $g^s \in \mathbb{Z}_p$ as a public key.

4.2.2. *Outsource.* When a user wants to store a file on the cloud storage, the interactive algorithm is run between them.

- (1) Given a data set $X = \{x_1, x_2, \dots, x_m\}$, the user uses an error correction code to get the encoded data Y . In the case that some blocks $Y' \subset Y$ may be lost by the cloud storage, an error correction code is used to reconstruct the original data set X [41].
- (2) Divide the encoded data Y into n blocks, $Y = \{y_1, y_2, \dots, y_n\}$, where $y_i \in \{0, 1\}^t$.
- (3) For each data block y_i , the user computes the authentication tag σ_i as follows:
 - (i) Randomly choose a nonzero element $r_i \in \mathbb{Z}_p^*$ called block nonce.
 - (ii) $\sigma_i = y_i^{H(r_i||i)}$
- (4) The user outsources Y and $\Sigma = \{\sigma_i | 1 \leq i \leq n\}$ to the storage server.
- (5) The cloud storage provider creates a Merkle tree $\text{Merkel}(\Sigma)$ by Σ , and stores the hash of the Merkle tree $H(\text{Merkel}(\Sigma))$ into the blockchain by doing an operation $\text{AppendBlock}(\text{Merkel}(\Sigma); ts, \text{NULL})$.

Remark 2. When we compute $y_i^{H(r_i||i)}$, y_i and $H(r_i||i)$ are treated as a big integer number.

4.2.3. *RequestChallenge.* To verify that the provider has stored the data correctly, the user randomly selects an integer $1 \leq k \leq n$ indicating which block should be checked. Then k and r_k are sent to the provider for requesting challenge.

4.2.4. *ResponseProof.* For the cloud storage provider, there are n blocks of data and the k -th block is requested to be checked. Now when the provider receives a request challenge, a PoR can be generated as follows:

- (1) Randomly select a nonzero element $x \in \mathbb{Z}_p$.
- (2) Compute $\sigma = \sigma_k^x$ and g^x .
- (3) Fetch out Y and Σ from storage devices to retain the hash of the Merkle tree $\text{Merkel}(\Sigma)$.
- (4) Send $H(\text{Merkel}(\Sigma))$ to the timestamp provider in the blockchain.

- (5) The timestamp provider verifies that $H(\text{Merkel}(\Sigma))$ is valid when received it. If it is valid, then a timestamp ts is generated to run

$$\text{AppendBlock}(\text{NewBlock}(\text{Merkel}(\Sigma), ts)), \quad (5)$$

and is sent back to the cloud storage provider. Otherwise, the algorithm is terminated.

- (6) The cloud storage provider generates the proof

$$\text{proof}_k = (\sigma, \sigma_k, g^x, ts, H_1, H_2), \quad (6)$$

where $H_1 = H(\text{Merkel}(\Sigma))$ and $H_2 = H(ts||\text{Merkel}(\Sigma))$. Then proof_k is sent back to the user.

4.2.5. *VerifyProof.* After receiving the proof, the user does the following operations in order:

- (1) Send ts to the timestamp provider in the blockchain. If no *accept* is returned, then the algorithm is terminated with a *reject*.
- (2) If the blockchain is invalid (See Section 2.3), then the algorithm is terminated with a *reject*.
- (3) Run $\text{FetchBlock}(ts)$ to obtain the corresponding Merkle tree $\text{Merkel}(\Sigma)$ and the hash H_0 of that Merkle tree from the blockchain.
- (4) If $H_0 \neq H(\text{Merkel}(\Sigma))$, then the algorithm is terminated with a *reject*.
- (5) If $\text{Merkel}(\Sigma)$ is invalid (See Section 2.2), then the algorithm is terminated with a *reject*.
- (6) If $H(\sigma_k)$ does not equal the value of the corresponding leaf node in the Merkle tree, then the algorithm is terminated with a *reject*.
- (7) If $e(\sigma, g^s) \neq e(\sigma_k^s, g^x)$, then the algorithm is terminated with a *reject*.
- (8) If $H_1 \neq H(\text{Merkel}(\Sigma))$, then the algorithm is terminated with a *reject*.
- (9) If $H_2 \neq H(ts||\text{Merkel}(\Sigma))$, then the algorithm is terminated with a *reject*.
- (10) Return *accept*.

Remark 3. Firstly, the above operations first check that the blockchain (without Merkle trees) and the Merkle tree related to the last block are valid. Secondly, the existence of the k -th block is checked.

5. Security Analysis

5.1. Correctness

Theorem 1. *The verify process is correct. It means that*

$$e(\sigma, g^s) = e(\sigma_k^s, g^x), \quad (7)$$

holds where $\sigma = \sigma_k^x$.

Proof. It follows from the property of bilinear pairing that

$$e(\sigma, g^s) = e(\sigma_k^x, g^s) = e(\sigma_k, g)^{sx} = e(\sigma_k^s, g^x). \quad (8)$$

□

Remark 4. Due to Assumptions 4 and 5, the private key s is still secure even the result of bilinear pairing computation is public.

5.2. Reasonableness

Theorem 2. *If the cloud storage provider is honest, the final proof must be*

$$\text{proof}_k = (\sigma, \sigma_k, g^x, ts, H_1, H_2), \quad (9)$$

where $H_1 = H(\text{Merkel}(\Sigma))$, $H_2 = H(ts \parallel \text{Merkel}(\Sigma))$ and ts is the current timestamp.

Proof. If the cloud storage provider is honest, the following points hold true:

- (i) σ and σ_k guarantee that at least the cloud storage provider stores the k -th block which is not revealed to the public in the bilinear pairing computation (See Remark 5.1).
- (ii) The Merkle tree is created by the cloud storage provider at the time ts was required, and the leaf nodes of the tree are all part of the data set Y . It follows from Assumptions 1 and 2 that these hashes cannot be found without knowing the original data set Y .
- (iii) ts generated by blockchain is trusted according to Assumption 3.
- (iv) The consistency of the Merkle tree and timestamp are assured by H_1 and H_2 , respectively.

To sum up, the cloud storage provider must store the data set correctly if *VerifyProof* return is *accepted*. □

5.3. Traceability

Theorem 3. *The blockchain-based PoR scheme in Section 4 is traceable.*

Proof. If the cloud server is dishonest, that is, the server modifies, deletes, or tampers with a piece of file without authorization of the user, S cannot compute the value of the root node correctly, so it cannot prove that he has completely stored the data. By verifying the Merkle tree, it will get which piece of file S has been modified finally.

For example, to verify whether the fifth block file has been modified, the following procedure can be followed and the structure as shown in Figure 4:

- (i) *Verify Node 1.* Verify that the calculated value of node 1 is correct through the values of node 2 and node 3.

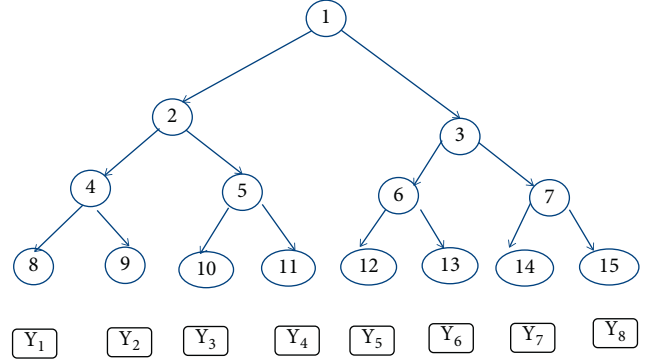


FIGURE 4: The example of traceability.

- (ii) *Verify the Value of Node 3.* The receiver computes the value of node 3 through the values of node 6 and node 7 that he has received and verifies whether the calculated value of node 3 is correct.
- (iii) *Compute the Value of Node 6.* The receiver computes the value of node 6 through the values of node 12 and node 13 that he has received and verifies whether the calculated value of node 6 is correct.
- (iv) The receiver computes the value of node 12 from the value of Y_5 and verifies that the calculated value of node 6 is correct.

The correct value can be determined by whether the value of node 6 is consistent. This allows you to track down blocks of files that have been modified. □

5.4. Resistance to Two Kinds of Attacks. In this subsection, two kinds of attacks are considered, i.e., replay attacks and collusion attacks.

5.4.1. Resistance to Replay Attack. In Section 4.2.4, note that there is a timestamp ts attached to the proof_k , where

$$\text{proof}_k = (\sigma, \sigma_k, g^x, ts, H(\text{Merkel}(\Sigma)), H(ts \parallel \text{Merkel}(\Sigma))). \quad (10)$$

- (i) If a data storage provider uses an old timestamp ts , then it would be *rejected* in the first step (1) in Section 4.2.5 since the timestamp provider can easily find that such ts is expired. In other words, such ts may be valid in a short time. However, the user could not run this protocol twice in such a short time.
- (ii) If a data storage provider uses an old proof proof_k , then it would be *rejected* in the seventh step (7) in Section 4.2.5 since g^x is attached with a challenge x that is randomly generated by the user. x should be different in two runs of this protocol.

In a word, our protocol is resistant to replay attacks with old timestamps ts or old proof proof_k .

5.4.2. Resistance to Collusion Attack. If we consider the case that the timestamp provider (and by extension the

blockchain provider) colludes with the data storage provider, then, in other words, the data storage provider would also play as a timestamp provider in the blockchain context. However, due to the security analysis of blockchain [42, 43], such malicious timestamp providers could be detected by the nodes in the blockchain network. Under Assumption 3 (BlockChain assumption), our protocol is resistant to such collusion attacks which can be reduced to an attack in a blockchain context.

6. Conclusion

In order to protect the security and integrity of user data, we formally defined a novel security model for a blockchain-based PoR scheme and proposed a secure scheme under the defined security model. The properties of the PoR scheme and the characteristics of blockchain, ensure the security and the integrity of data, respectively. Furthermore, we prove the correctness and reasonableness of our scheme. Our scheme makes user data more secure. In our scheme, blockchain plays an irreplaceable role in the privacy and security of user data. It is believed that as a blockchain improves the PoR scheme, it will continue to promote the progress of technology.

However, there are still many attacks not being considered, such as reset attacks and malicious attacks. To improve the performance, it is interesting to remove the bilinear mapping while reserving the same security level.

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 in this work.

Acknowledgments

This work was supported by the Higher Education Technology Innovation Projects Foundation of Shanxi (Grant nos. 2021L467 and 2020L0560), National Statistical Science Research Program of China (Grant no. 2021LY047), Research Project of Yuncheng University (Grant nos. YQ-2020020 and XK-2020036), Key Discipline Project of Yuncheng University, Young Innovative Talents Project of General Colleges and Universities in Guangdong Province (Grant no. 2019KQNCX112), Talent Special Project of Research Project of Guangdong Polytechnic Normal University (Grant no. 2021SDKYA051), Opening Project of Guangdong Provincial Key Laboratory of Information Security Technology (Grant no. 2020B1212060078), and the Guangdong Basic and Applied Basic Research Foundation (Grant no. 2021A1515011954).

References

- [1] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [2] R. Buyya, C. Shin Yeo, S. Venugopal, B. James, and I. Brandic, "Cloud computing and emerging it platforms: vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.
- [3] L. M. Kaufman, "Data security in the world of cloud computing," *IEEE Security & Privacy*, vol. 7, no. 4, pp. 61–64, 2009.
- [4] R. L. Grossman, "The case for cloud computing," *IT professional*, vol. 11, no. 2, pp. 23–27, 2009.
- [5] C. Stergiou, K. E. Psannis, B. G. Kim, and B. Gupta, "Secure integration of iot and cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 964–975, 2018.
- [6] A. Mishra, N. Gupta, and B. B. Gupta, "Defense mechanisms against ddos attack based on entropy in sdn-cloud using pox controller," *Telecommunication Systems*, vol. 77, pp. 1–16, 2021.
- [7] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "Psspr: a source location privacy protection scheme based on sector phantom routing in wsns," *International Journal of Intelligent Systems*, vol. 37, no. 2, 2021.
- [8] D. Zisis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.
- [9] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications*, vol. 1, no. 1, pp. 7–18, 2010.
- [10] Y. Jadeja and K. Modi, "Cloud computing-concepts, architecture and challenges," in *Proceedings of the International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*, pp. 877–880, Nagercoil, India, March 2012.
- [11] C. B. Tan, M. H. A. Hijazi, Y. Lim, and A. Gani, "A survey on proof of retrievability for cloud data integrity and availability: cloud storage state-of-the-art, issues, solutions and future trends," *Journal of Network and Computer Applications*, vol. 110, pp. 75–86, 2018.
- [12] L. Jin, X. Tan, X. Chen, D. S. Wong, and F. Xhafa, "Opor: enabling proof of retrievability in cloud computing with resource-constrained devices," *IEEE Transactions on cloud computing*, vol. 3, no. 2, pp. 195–205, 2014.
- [13] Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in *Proceedings of the Theory of Cryptography Conference*, pp. 109–127, CA, USA, March 2009.
- [14] H. Shacham and B. Waters, "Compact proofs of retrievability," *Journal of Cryptology*, vol. 26, no. 3, pp. 442–483, 2013.
- [15] K. D. Bowers, A. Juels, and A. Oprea, "Hail: a high-availability and integrity layer for cloud storage," in *Proceedings of the 16th ACM conference on Computer and communications security*, pp. 187–198, IL, USA, November 2009.
- [16] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote data checking for network coding-based distributed storage systems," in *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, pp. 31–42, IL, USA, October 2010.
- [17] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "Mr-pdp: multiple-replica provable data possession," in *Proceedings of the 28th international conference on distributed computing systems*, pp. 411–420, Beijing, China, July 2008.

- [18] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proceedings of the 4th international conference on Security and privacy in communication networks*, Istanbul Turkey, September 2008.
- [19] M. Bellare and O. Goldreich, "On defining proofs of knowledge," in *Proceedings of the Annual International Cryptology Conference*, pp. 390–420, August 1992.
- [20] Q. Wang, C. Wang, L. Jin, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *Proceedings of the European symposium on research in computer security*, pp. 355–370, Saint-Malo, France, September 2009.
- [21] M. Blum, W. Evans, P. Gemmell, S. Kannan, and M. Naor, "Checking the correctness of memories," *Algorithmica*, vol. 12, no. 2-3, pp. 225–244, 1994.
- [22] A. Fu, Y. Li, S. Yu, Y. Yan, and G. Zhang, "Dipor: an ida-based dynamic proof of retrievability scheme for cloud storage systems," *Journal of Network and Computer Applications*, vol. 104, pp. 97–106, 2018.
- [23] D. Cash, A. K p c , and D. Wichs, "Dynamic proofs of retrievability via oblivious ram," *Journal of Cryptology*, vol. 30, no. 1, pp. 22–57, 2017.
- [24] M. Etemad and A. K p c , "Transparent, distributed, and replicated dynamic provable data possession," in *Proceedings of the International Conference on Applied Cryptography and Network Security*, Saint-Malo, France, June 2013.
- [25] E. Stefanov, M. V. Dijk, A. Juels, and A. Oprea, "Iris: A scalable cloud file system with efficient integrity checks," in *Proceedings of the 28th Annual Computer Security Applications Conference*, pp. 229–238, Orlando, FL, USA, January 2011.
- [26] E. Staff, "Blockchains: the great chain of being sure about things," *Economist*, vol. 18, no. 7, 2016.
- [27] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Decentralized Business Review*, Article ID 21260, 2008, <https://www.debr.io/article/21260>.
- [28] A. C. Chow, M. W. C. Paul, P. A. J. Haldenby et al., "Automated implementation of provisioned services based on captured sensor data," US Patent App, 2018.
- [29] C. Xie, Y. Sun, and H. Luo, "Secured data storage scheme based on block chain for agricultural products tracking," in *Proceedings of the Third International Conference on Big Data Computing and Communications (BIGCOM)*, pp. 45–50, Chengdu, China, August 2017.
- [30] H. Y. Chen, "An e-government model design based on block chain," in *Proceedings of the International Conference on Manufacturing, Construction and Energy Engineering*, Hong Kong, China, November 2017.
- [31] A. Kiayias, A. Russell, B. David, and R. O. Ouroboros, "A provably secure proof-of-stake blockchain protocol," in *Proceedings of the Annual International Cryptology Conference*, pp. 357–388, Santa Barbara, USA, July 2017.
- [32] T. Li, Y. Chen, Y. Wang et al., "Rational protocols and attacks in blockchain system," *Security and Communication Networks*, vol. 2020, Article ID 8839047, 11 pages, 2020.
- [33] Y. Wang, G. Yang, T. Li et al., "Optimal mixed block withholding attacks based on reinforcement learning," *International Journal of Intelligent Systems*, vol. 35, no. 12, pp. 2032–2048, 2020.
- [34] C. Wang, K. Ren, W. Lou, and L. Jin, "Toward publicly auditable secure cloud data storage services," *IEEE network*, vol. 24, no. 4, pp. 19–24, 2010.
- [35] A. Juels and B. S. Kaliski, "Pors: proofs of retrievability for large files," in *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 584–597, Alexandria, VA, USA, October 2007.
- [36] "Wikipedia contributors. Comparison of cryptographic hash functions," 2021, https://en.wikipedia.org/wiki/Comparison_of_cryptographic_hash_functions.
- [37] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
- [38] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, 2021.
- [39] X. Yu, Y. W. Zhaojie, F. Li et al., "Impsuic: a quality updating rule in mixing coins with maximum utilities," *International Journal of Intelligent Systems*, vol. 36, no. 3, pp. 1182–1198, 2020.
- [40] J. Antoine, "A one round protocol for tripartite diffie-hellman," *Journal of Cryptology*, vol. 17, no. 4, pp. 263–276, 2004.
- [41] A. Juels, J. Kelley, R. Tamassia, and N. Triandopoulos, "Falcon codes: fast, authenticated It codes (or: making rapid tornadoes unstoppable)," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1032–1047, Denver, CO, USA, October 2015.
- [42] A. Kircanski and T. Tarvis, "Coinbugs: enumerating common blockchain implementation-level vulnerabilities," 2021, <https://arxiv.org/abs/2104.06540>.
- [43] D. Goldman, "The Verge Hack," explained, 2018, <https://blog.theabacus.io/the-verge-hack-explained-7942f63a3017>.

Research Article

An Android Malicious Application Detection Method with Decision Mechanism in the Operating Environment of Blockchain

Xingyu Li , Zongqu Zhao , Yongli Tang , Jing Zhang , Chengyi Wu , and Ying Li 

Henan Polytechnic University, School of Computer Science and Technology, Jiaozuo, Henan 454000, China

Correspondence should be addressed to Yongli Tang; yltang@hpu.edu.cn

Received 6 November 2021; Revised 27 November 2021; Accepted 11 January 2022; Published 29 January 2022

Academic Editor: Yuling Chen

Copyright © 2022 Xingyu Li 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, security policies and behaviour detection methods have been proposed to improve the security of blockchain by many researchers. However, these methods cannot discover the source of typical behaviours, such as the malicious applications in the blockchain environment. Android application is an important part of the blockchain operating environment, and machine learning-based Android malware application detection method is significant for blockchain user security. The way of constructing features in these methods determines the performance. The single-feature mechanism, training classifiers with one type of features, cannot detect the malicious applications effectively which exhibit the typical behaviours in various forms. The multifeatures fusion mechanism, constructing mixed features from multiple types of data sources, can cover more kinds of information. However, different types of data sources will interfere with each other in the mixed features constructed by this mechanism. That limits the performance of the model. In order to improve the detection performance of Android malicious applications in complex scenarios, we propose an Android malicious application detection method which includes parallel feature processing and decision mechanism. Our method uses RGB image visualization technology to construct three types of RGB image which are utilized to train different classifiers, respectively, and a decision mechanism is designed to fuse the outputs of subclassifiers through weight analysis. This approach simultaneously extracts different types of features, which preserve application information comprehensively. Different classifiers are trained by these features to guarantee independence of each feature and classifier. On this basis, a comprehensive analysis of many methods is performed on the Android malware dataset, and the results show that our method has better efficiency and adaptability than others.

1. Introduction

Blockchain is a new decentralized infrastructure and distributed computing paradigm emerging with the increasing popularity of digital cryptocurrencies such as Bitcoin. In recent years, many researchers have put forward a large number of researches on the security of blockchain from two aspects: security policy and behaviour detection. The security policies include a series of technologies such as privacy protection [1, 2] and data encryption [3]. Behaviour detections are mainly used to identify typical behaviours that existed in blockchain, such as mining [4, 5]. However, the malicious applications which result in these behaviours cannot be discovered by security policy and behaviour detection method. Android is a free and open-source operating system based on Linux, widely used in blockchain.

Due to Android's openness, it has become a main target of malicious applications. The types of new Android malicious applications are cost consumption, privacy theft, remote control, roguery, malicious deduction of fees, fraud, etc. Detecting malicious Android applications is greatly significant for improving the security of Android applications and protecting the blockchain users' secret keys and information security.

In order to effectively detect Android malicious applications, various methods have been proposed. These methods include the single-feature mechanisms [6–17] and the multifeatures fusion mechanisms [18–26]. The methods based on a single-feature mechanism usually train a classifier with one type of features which include APIs [8], permissions [6, 7], call graphs, images [10], or codes [9, 13–17]. As the structural complexity of these features increases, the

accuracy of methods is constantly improved. However, with the development of the malwares, the malicious behaviours would be presented in multiple places in the application simultaneously, such as signature files, configuration files, code files, DLL files, and so on. A classifier trained by a single-feature cannot detect various malwares effectively. Therefore, it is difficult to improve the performance of the method based on a single-feature mechanism when detecting various Android malicious applications. The multifeatures fusion mechanisms can solve the problems encountered by a single-feature mechanism. These mechanisms construct a group of hybrid features which include multiple types of data. Then they use hybrid features to train the classifiers. To some extent, these mechanisms could detect those malicious applications which present typical behaviours in multiple types of data sources. However, as multiple types of data sources are fused into a group of hybrid features, different types of data sources can interfere with each other during the classifier training process, which restricts the further improvement of the performance.

We propose an Android malicious applications detection method including a decision mechanism. The method introduces a concept of the parallel detection and fusing detection result. The parallel detection constructs a variety of images for different types of data sources to train multiple classifiers. The result decision layer is constructed to fuse the outputs of multiple classifiers. The advantages of our method are as follows:

- (i) Different types of images are constructed using parallel RGB image visualization techniques, and these images are employed to train multiple classifiers. These mutual independent classifiers can reduce the interference between different data sources.
- (ii) The result decision layer fuses the outputs of multiple classifiers with decision algorithms. It guarantees the independence and accuracy of subclassifiers and improves the performance of the primary classifier.

2. Related Work

In this section, we will introduce Android malicious application detection methods under different mechanisms in detail.

2.1. The Methods Based on Single-Feature Mechanism. The methods based on single-feature mechanism usually train the classifier with one type feature. Almin and others [6] extracted the permission as feature from applications. Then, they adopted k-means algorithm to cluster these permissions. A detection method was developed by Li et al. [7]. They named it Significant Permission Identification (SIGPID) which identified an essential subset of permissions with three types of data analysis: permission ranking with negative rate, support based permission ranking, and permission mining with association rules. SIGPID trained SVM classifiers with the essential subset of permissions. Their method reduced the number of permissions that need to be analysed. Chen et al. [8] trained machine learning classifiers

using APIs feature extracted from the smali files. To improve the robustness of online malware detectors, they proposed a robust secure-learning paradigm. Zhang et al. [9] converted the opcode sequences into an image and finally performed further feature extraction using CNN. Nataraj et al. [10] mapped malicious applications to grayscale images firstly and then obtained features through the Gabor filter. Lin et al. [11] extracted the system call sequence of an app at run-time. Then they used subsequences to detect some special Android malware generated by piggybacking malicious payloads into benign applications. Munoz and others [12] selected predictive features from the metadata which was collected from Google play. Dixon and others [13] detected malware code behaviour by using the power consumption feature based on time and location. Karbab and others [14] proposed Mal-Dozer, an automatic Android malware detection and family attribution framework that relies on sequence classification using deep learning techniques. Canfora and others [15] characterized the frequencies of opcode N -grams and used the Random Forest classifier to test detection accuracy under different values of N . In order to get the best detection results, they carried out many tests under different parameters. The average detection rate reached 97%. Li et al. [16] extracted the number of opcodes in each sample into a binary matrix of the same size. Then they took advantage of binary matrices to train CNN. This detection system achieved an accuracy of 99%. Zhang et al. [17] calculated the n -gram value of the opcode. The value of the opcode was divided into SA-CNN slices to train CNN. The shape of every SA-CNN was (M, N) . The result showed that the experimental index was optimal when (M, N) was $(400, 10)$. Compared with Canfora and others [15], Li et al. [16] and Zhang et al. [17] avoided the tedious process of selecting parameters. Most of the parameters of CNN could be obtained through feature training process.

2.2. The Methods Based on Multifeatures Fusion. The methods based on multifeatures fusion are a concept of early fusion. These methods usually fuse multiple single features into a group of mixed features. The mixed features are used to train a traditional machine learning algorithm or a deep learning algorithm. Peiravian and others [18] used the call relationships between function packages and classes in the applications as a feature which could present APIs. They also got the permission application list from configuration files. This list was another feature which could present permissions. Then they fused these two classes of features into a feature set and used them to train traditional machine learning classifiers. The classifiers could detect malicious behaviours in Android applications. Afonso and others [19] constructed a group of hybrid features including API calls and system call traces. For classification, Random Forest classifier was utilized. Arp et al. [20] proposed a lightweight detection method, which was named Drebin. The Drebin classifier was trained by a group of mixed features which were made up of permissions and APIs. This method significantly enhanced the detection ability and efficiency. Zhang et al. [21] used binary values to represent opcodes,

permissions, and API usage frequency values in the application. The binary was converted into a RGB image as a feature. Deep learning algorithm could obtain rules hidden in the data by learning sample data. Han et al. [22] proposed a hybrid feature construction method named MalDAE that fused the dynamic and static API sequences. Feng et al. [23] fused manifest properties and API calls into a hybrid matrix; this matrix was the training feature of deep neural networks. Arshad and others [24] explained a 3-level hybrid malware detection model named SAMADroid. They extracted dynamic feature in level 1 and static feature in level 2 and trained machine learning classifier using dynamic and static feature in level 3. Suarez-Tangil and others [25] proposed the DroidSieve method from which several static features were extracted. These features included permissions, APIs, and application components. Holland et al. [26] and Quan et al. [27] adopted pattern match algorithm and the mixed feature to detect malwares.

2.3. The Methods Based on Decision Mechanism. The methods utilizing decision mechanisms usually train multiple detection classifiers with single or multiple features. The results of the multiple classifiers are then fused into a final detection result using a decision algorithm or decision layer. Wu et al. [28] extracted multiple types of features from the application, which included permissions, subassemblies, the information of intent, and APIs. These features were used separately to train different classifiers using traditional machine learning algorithms. The detection results of these classifiers were combined in pairs to make a decision. The decision result was taken as the final test result. Tang et al. [29] extracted two types of features: opcode n-gram and the frequencies of duplicate code subblocks. They trained the XGBoost classifier with opcode n-gram and the Random Forest classifier with the frequencies of duplicate code subblocks. Then they added a decision algorithm after these two classifiers. Ananya et al. [30] used dynamic analysis to get application system calls, which was represented by the n-gram algorithm. The n-gram of the system calls was utilized as a feature to train a machine learning classifier and a DNN classifier separately. Finally, they adopted a decision algorithm to fuse the results of the two classifiers.

2.4. Summary and Analysis. With the development of malicious application technologies, the variability and uncertainty of the new type malicious applications greatly reduce the performance of the detection methods based on single-feature mechanism. The methods based on multi-features fusion mechanism, which fuses multiple types of features into a set of tensors to train one type of classifier, solve the problem of single-feature mechanisms. They enriched the information in the features. The method based on multi-features fusion mechanism could stably detect the polytropic malicious behaviours in applications. However, with the number of type features increasing, different types of features in the mixed tensor interfere with each other when training the classifier. It will increase the false positive rate for benign applications, reduce the recall rate for benign

applications, and limit further improvement of the detection capability.

Based on the above problems, we propose an Android malicious application detection method with a decision mechanism. Our method uses feature construction methods and detection algorithms which show good performance in single-feature mechanism and multi-features fusion mechanism. Based on this, we enhance the performance. The features constructed by these methods can be divided into self-defined structured features and image features. Self-defined structure features include APIs, permissions, opcode, system calls, etc. It usually relies on the disassembly techniques when constructing these features. The accuracy and comprehensiveness of these features will be disturbed by shell and code obfuscation techniques which are often used to prevent the analysis of the application by external programs or software. Image features can be constructed with RGB image visualization technologies. These technologies convert binary files directly into RGB images. It avoids some of the detection problems of classifiers, which are caused by disassembly applications.

The classifiers of technologies can be classified into traditional machine learning classifiers and deep learning classifiers. Compared to traditional machine learning classifiers, the parameters and weights of deep learning classifiers could be obtained through a self-study process. The ability of the deep learning classifiers is more stable.

3. Our Approach

The Android malicious application detection model, shown in Figure 1, consists of three processes: parallel RGB image visualization, parallel detection classifier, and decision layer. The process of parallel RGB image visualization constructs three types of images which are utilized to train the detection classifiers separately. The process of parallel detection classifier consists of three separate classifiers which adopt the VGG16 algorithm. The purpose of the process decision layer is to fuse the outputs of the parallel detection classifier by using a decision algorithm.

3.1. Parallel RGB Image Visualization Technology. As shown in Figure 2, the parallel RGB image visualization technology will create three types of images: dex-image, manifest-image, and certificates-image. Dex file is the data source for dex-image, which contains all compiled Java code. Android manifest file is the data source for the manifest-image. It is usually stored in the root directory. The certificate files mainly contain MF, SF, and RSA files. They can be regarded as the containers of APK to record the digest information of all files in APK. These files are the data source of certificates-image.

These images are created by the image visualization technology. We get the dex files, the Android manifest files, and the signature files in the META-INF folder by unpacking the APK. Then we extract three binary strings, B_{dex} , $B_{manifest}$, $B_{certificates}$, from these files. B_i is a string

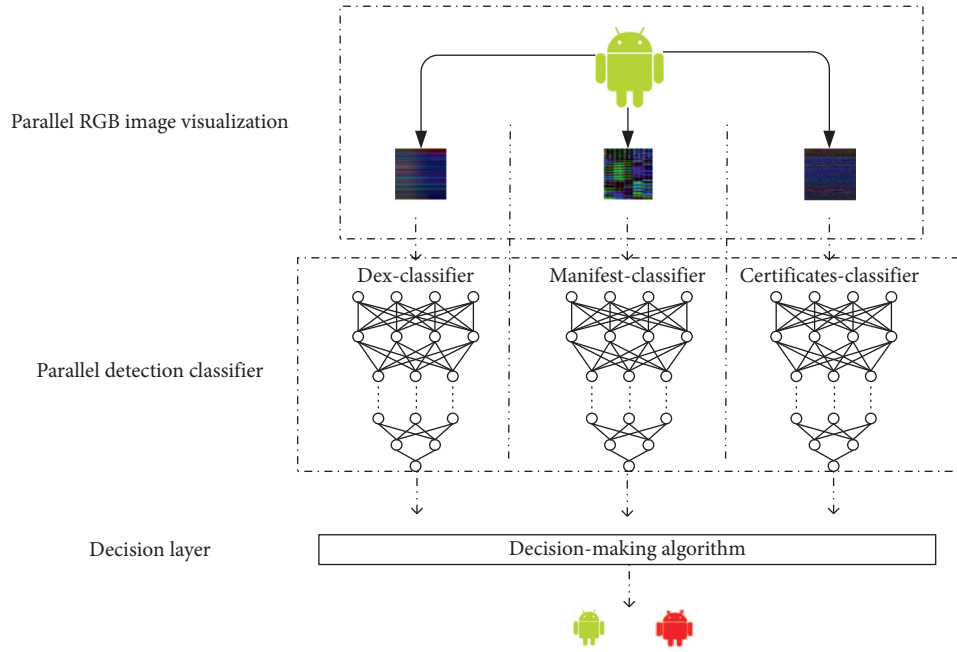


FIGURE 1: Android malicious application detection model diagram.

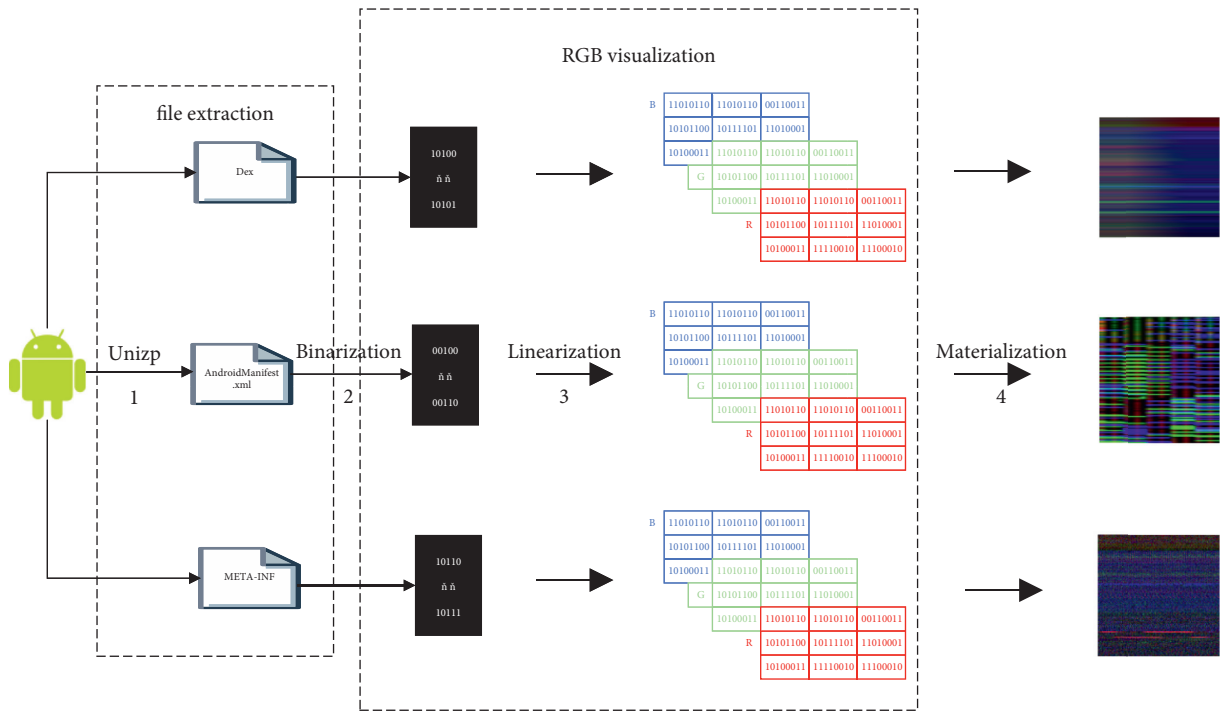


FIGURE 2: Parallel RGB image visualization technology.

containing 0 and 1. The RGB visualization technology is to convert B_i into a “red-green-blue” image. Each pixel is composed of three channels. The value range of each channel is $[0, 255]$. RGB visualization requires the following three basic steps.

(i) Divide B_i into three equal segments of characteristic binary code of the same length b_{ir}, b_{ig}, b_{ib} . b_{ir} is the binary code snippet for the red channel. b_{ig} is the

binary code snippet for the green channel. b_{ib} is the binary code snippet for the blue channel.

(ii) Divide each code segment into subcode segments of 8-bit length. Each subcode segment represents the value of each channel, p_i , at the pixel point. So, $b_{ir} = \{p_{ir1}, p_{ir2}, \dots, p_{irj}\}$, $b_{ig} = \{p_{ig1}, p_{ig2}, \dots, p_{igj}\}$, $b_{ib} = \{p_{ib1}, p_{ib2}, \dots, p_{ibj}\}$. Suppose a binary code segment is 0101011011010111. This

process is 0101011011010111 \rightarrow
01010110, 11010111 \rightarrow 86, 215.

- (iii) Finally, transform b_{ir}, b_{ig}, b_{ib} into the pixel matrix of column J and row K . Transform b_{ir}, b_{ig}, b_{ib} into three matrices whose dimensions are $[J, K]$. Algorithm 1 is the way to get the column J and row K . Then pad the data of b_i into the matrix M_i whose dimension is $[J, K]$. Finally, use the Image.fromarray function to convert $[M_{ir}, M_{ig}, M_{ib}]$ into a RGB image.

3.2. Parallel Detection Classifier. In this process, we choose VGG16 neural network algorithm developed in 2014 to generate three classifiers: dex-classifier, manifest-classifier, and certificates-classifier. The VGG16 neural network algorithm has 16 parameter layers and 5 no-parameter layers: 13 convolutional layers, 3 fully connected layers, and 5 maximum pooling layers.

Convolution layer: The convolution layer consists of several convolution kernels. Formulas (1) and (2) are the calculation equations of the convolution layer.

$$N = \frac{(W - F + 2P)}{S + 1}, \quad (1)$$

$$c_j = f_1(\text{Conv}(\mathbf{M}, \mathbf{w}_j) + \mathbf{b}_j). \quad (2)$$

Max-pooling layer: The purpose of the Max-pooling layer is to extract the maximum value of the target region. The filter size is 2×2 . Stride is 2.

Fully connected layer: Before the fully connected layer, the output matrix of the last pooling needs to be stretched into a one-dimensional vector \mathbf{z} by a flattening function. The output of the previous fully connected layer is the input of the next fully connected layer. Each node in the fully connection layer is connected to all nodes in the preceding layer. Formula (3) is the calculation formula of the fully connected layer.

$$\mathbf{y} = f_2(\mathbf{W}_f \cdot \mathbf{z}) + \mathbf{b}'_f. \quad (3)$$

3.3. Decision Layer. The outputs of three classifiers are 0 or 1, respectively. When the output of a classifier is 0, the classifier considers that such data sources cannot present typical behaviours. When the output of a classifier is 1, the classifier considers that such data sources can present typical behaviours. For example, if the typical behaviours are presented by the dex file, the detection value of dex-classifier is 1. In the process of parallel detection classifier, each application has three detection values which are the outputs of dex-classifier, manifest-classifier, and certificates-classifier. Then, we use the decision algorithm to fuse the three detection values. The output of decision algorithm is the predicted value for each application. When the result of any classifier is 1, the decision algorithm considers the application as a malicious application. For example, the outputs of the three classifiers are (x, y, z) . x is the result of dex-classifier. y is the result of manifest-classifier. z is the result of certificates-classifier. If $x == 1 | y == 1 | z == 1, o = 1$. The

application is malware. o is the output of the decision algorithm. If $x = 1 \& y = 1 \& z = 1, o = 0$. The application is a benign application. Table 1 shows the results of all decisions algorithm.

4. Experiments

In order to evaluate detection techniques with decision mechanisms, we conduct experiments for stability evaluation in three datasets: AndMal2017, CICMalDroid2020, and DREBIN. Then, we verify the effectiveness of decision mechanism by comparing the detection results of dex-classifier, manifest-classifier, certificates-classifier, and decision mechanism. In the end, we compare the detection results under the three mechanisms: decision, single-feature, multifeatures fusion.

4.1. Environment and Datasets. The equipment used in our experiment is a machine with 32G RAM, 1T HDD, and Intel(R) Xeon(R) Silver 4214 CPU operating at 2.20 GHz. Table 2 shows the three datasets for our experiments.

4.2. Evaluation Parameters. In order to evaluate the effectiveness of our proposed method, we adopt some evaluation parameters, including precision, accuracy, TPR, f1-score, receiver operating characteristic (ROC) curve, and Area Under Curve (AUC). These parameters help us to evaluate the effectiveness of our method. TP is the number of applications correctly classified as malicious. FP is the number of benign applications incorrectly classified as malicious. TN is the number of benign applications correctly classified as benign. FN is the number of malicious applications incorrectly classified as benign.

Precision is defined as

$$\text{Precision} = \frac{\text{TN}}{\text{FN} + \text{TN}} / \frac{\text{TP}}{\text{FP} + \text{TP}}. \quad (4)$$

Accuracy can be calculated by

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}}. \quad (5)$$

TPR, also known as recall rate, is defined as

$$\text{TPR (Recall)} = \frac{\text{TN}}{\text{TN} + \text{FP}} / \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (6)$$

Naturally, F1-score is defined as

$$\text{F1 - score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}. \quad (7)$$

The ROC curve is called sensitivity curve. All points on the curve reflect the same sensitivity, which is the result of the response to the same signal stimulus under several different criteria. AUC is the area enclosed by ROC curve and coordinate axes.

4.3. The Detection Effect of Three Datasets. Table 3 presents the results of the decision mechanisms utilized in AndMal2017, CICMalDroid2020, and DREBIN datasets. As

```

(i) Input  $b_{ij}$ ; Output:  $J&K$ 
 $l = \text{length}(b_{ij})$   $J = \text{sqrt}(l)$  If  $l \% J == 0$ :  $K = l // J$ 
Else: low = pow( $J, 2$ ) high = pow( $J + 1, 2$ )
If
( $l - \text{low}$ )  $\geq$  ( $\text{high} - l$ ):  $J = J + 1$   $J = J - 1$ 
While  $J = J - 1$ :
If ( $l \% J$ ): break  $J - = 1$   $K = l // J$ 

```

ALGORITHM 1: Getting the column J and row K

TABLE 1: Decision algorithm.

Dex-classifier	Manifest-classifier	Certificates-classifier	Decision result
0	0	0	0
0	1	0	1
0	0	1	1
0	1	1	1
1	0	0	1
1	1	0	1
1	0	1	1
1	1	1	1

TABLE 2: Datasets used in the experiment.

	AndMal2017	CICMalDroid2020	DREBIN
Benign application	1000	2000	2000
Malicious applications	500	500	500

TABLE 3: The detection effect of three datasets%.

Datasets		AndMal2017	CICMalDroid2020	DREBIN
TPR	Benign	96	94	96
	Malicious	89	87	84
F1-score	Benign	95	95	92
	Malicious	88	83	91
Precision	Benign	96	97	98
	Malicious	87	85	90
Accuracy		94	92	93

shown in Table 3, the TPR, f-score, and precision of malicious applications are all between 85% and 90%. The TPR, f-score, and precision of benign applications are all between 95% and 98%. The accuracy of the datasets is 94%, 92%, and 93%, respectively.

Figure 3 is the ROC curve for the three datasets, which shows the AUC curve and ROC values for the three datasets. The AUC values are 0.92, 0.88, and 0.87.

As shown in the experimental data in Table 3, the evaluation parameters of the three datasets fluctuate within a small range. There are no extreme differences in the experimental data due to the variation of the datasets. This indicates that our method has good detection stability. In different complex scenarios, the Android malicious application detection method with a decision mechanism can maintain excellent performance.

4.4. The Detection Results of Primary Classifier and Subclassifiers. The primary classifier is the model shown in Figure 1. The subclassifiers are the three classifiers before the decision layer: dex-classifier, manifest-classifier, and certificates-classifier.

The detection results of primary classifier and subclassifiers on dataset AndMal2017 are shown in Table 4. The malicious' TPR of the subclassifiers is between 40% and 50%. The malicious' F1-score of subclassifiers is between 55% and 65%. The malicious' precision of subclassifiers is between 80% and 90%. All the benign applications' evaluation parameters of subclassifiers are above 85%. All the malicious' evaluation parameters of primary classifier are above 85%. All the benign applications' evaluation parameters of primary classifier are above 95%. The accuracy rates of the primary classifier and subclassifiers are 94%, 88%, 82%, and

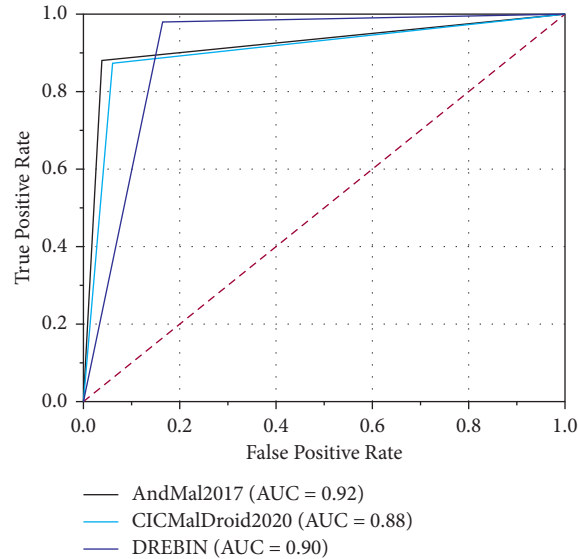


FIGURE 3: The ROC curve of three datasets.

TABLE 4: The evaluation parameters of primary classifier and subclassifiers%.

Classifiers		Primary	Dex-classifier	Manifest-classifier	Certificates-classifier
TPR	Benign	96	99	93	98
	Malicious	89	48	40	42
F1-score	Benign	95	93	89	92
	Malicious	88	62	58	56
Precision	Benign	96	88	86	87
	Malicious	87	89	80	85
Accuracy		94	88	82	86

86%, respectively. The AUC values, shown in Figure 4, are 0.92, 0.73, 0.67, and 0.70.

According to the experimental data in Table 4, subclassifiers and the primary classifier have similar detection performance for benign applications. However, the evaluation parameters of the primary classifier for malicious applications are twice that of the subclassifier. According to the needs of application developers, the malicious behaviour of malwares is mainly distributed in dex files, configuration files, and dynamic link library files. Each subclassifier of the primary classifier can detect malicious behaviour in only one class of files. The following is an example of using the dex-classifier. When the typical behaviours are presented by the Java code, the dex-classifier can accurately detect malicious. Based on multiple subclassifiers, the primary classifier containing decision algorithms makes up for the inability of subclassifiers to detect the multiple type sources of the typical behaviours. The primary classifier first uses subclassifiers to the sources of the typical behaviours in various files. In this process, the parallel detection of subclassifiers does not interfere with each other. The outputs of the subclassifiers are then fused with the decision algorithm. The primary classifier identifies applications with maliciousness detected by any subclassifier as malware. The primary classifier increases the weight of detecting malicious

applications. As a result, the detection performance of malicious applications has been significantly improved.

4.5. The Effectiveness under Different Mechanisms. We use different detection mechanisms to conduct experiments on the dataset AndMal2017. Table 5 shows the results. The TPR, f1-score, and precision of the single-feature mechanism against malicious applications are 63%, 69%, and 70%. The TPR, f1-score, precision of the multifeatures fusion mechanism against malicious applications are 83%, 76%, and 71%. The TPR, f1-score, and precision of the decision mechanism against malicious applications are 89%, 88%, and 87%. The TPR, f1-score, and precision of the single-feature mechanism against benign applications are 95%, 93%, and 90%. The TPR, f1-score, and precision of the multifeatures fusion mechanism against benign applications are 91%, 93%, and 95%. The TPR, f1-score, and precision of the decision mechanism against benign applications are 96%, 95%, and 96%. The accuracy of decision mechanism, single-feature mechanism, and multifeatures fusion mechanism is 94%, 85%, and 89%, respectively. The AUC values, as shown in Figure 5, are 0.92, 0.78, and 0.87.

As shown in the experimental data in Table 5, the detection ability of the multifeatures fusion mechanism is better than that of the single-feature mechanism. In essence,

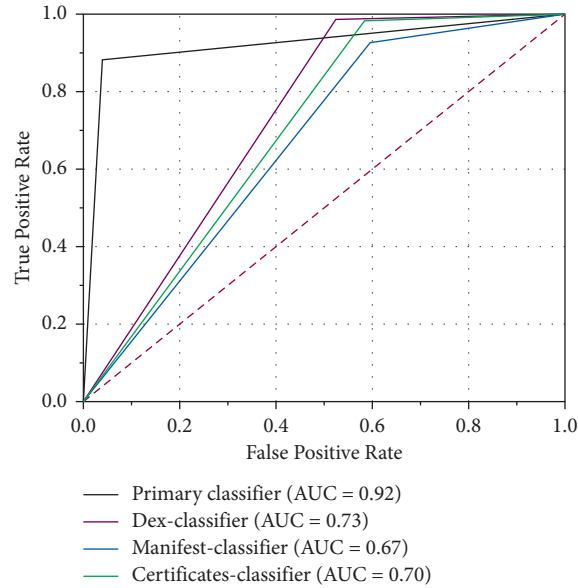


FIGURE 4: The ROC curve of primary classifier and subclassifiers.

TABLE 5: The evaluation parameters of different mechanisms%.

Detection mechanisms		Decision	Single-feature [7]	Multifeatures fusion [21]
TPR	Benign	96	95	91
	Malicious	89	63	83
F1-score	Benign	95	93	93
	Malicious	88	69	76
Precision	Benign	96	90	95
	Malicious	87	70	71
Accuracy		94	85	89

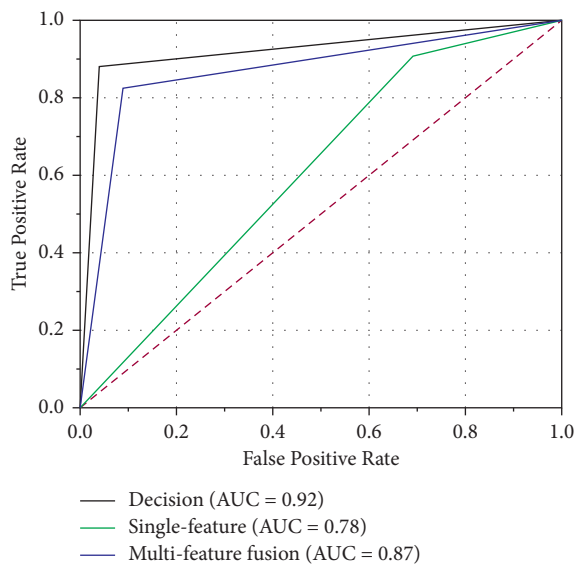


FIGURE 5: The ROC curve of different mechanisms.

the single-feature mechanism could only detect malicious behaviour represented by one type of feature. Take for example the sensitive permission features used by Li et al. [7].

Their method can only detect malicious behaviour for sensitive permissions. Their methods cannot effectively detect those applications which implement malicious behaviours through Java code, API calls, and dynamic link libraries. The multifeatures fusion mechanism is an improvement on the single-feature mechanism, which fuses many different types into one tensor feature. Tensor features include many types of features, such as code features, APIs, and permissions. Similar to the multifeatures fusion mechanism, our method converts different types of files into images. Compared with the single-feature mechanism, the multiple images used in our method and the mixed features used in the multifeatures fusion mechanism contain more abundant and comprehensive information. Therefore, the detection capability of our method and multifeatures fusion mechanism for malicious applications is much higher than that of the single-feature mechanism.

However, the recall rate of multifeatures fusion mechanism for benign applications is weaker than that of single-feature mechanism and our proposed method. The multifeatures fusion mechanism fuses multiple types of features into a set of tensors with fixed dimensions for training one type of classifier. Different types of features fused in a set of tensors will interfere with each other. It reduces the performance of the classifier for benign applications and limits

the improvement of the overall performance of the classifier. Our method separates different data sources and trains different subclassifiers by adopting parallel detection method. This approach avoids the problem of interference between data sources. Finally, the decision algorithm is used to fuse the detection results of multiple subclassifiers and improve the performance of the main classifier.

5. Conclusion

We propose an Android malicious application detection method which includes a decision mechanism to enhance the security of the blockchain operating environment. It improves the performance of Android malicious applications in complex scenarios. The parallel RGB image visualization technology in our method constructs three types of RGB images from dex files, manifest files, and certificates files. These images are then used to train three types of classifiers, respectively. Such technology reduces the interference between different data sources. Furthermore, we adopt a decision mechanism which adds a decision layer to the subclassifiers. The decision layer improves the performance by fusing the results of the three subclassifiers. Although our approach will be costly when loading multiple subclassifiers simultaneously, the efficiency of our scheme would be improved, if the subclassifiers can be deployed on different cloud detection servers.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This work was supported by the Research Foundation of Support Plan of Scientific and Technological Innovation Team in Universities of Henan Province 20IRTSTHN013, Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications, Xi'an, Shaanxi 710121, China, ICNS202006, the Fundamental Research Funds for the Universities of Henan Province, NSFRF210312, Youth Talent Support Program of Henan Association for Science and Technology, 2021HYTP008, and Project supported by the PhD Foundation of Henan Polytechnic University, B2021-41.

References

- [1] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1204–1221, 2022.
- [2] C. Song, Y. Zhang, X. Gu, L. Wang, and Z. Liu, "A trajectory substitution privacy protection scheme in location-based services," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 13, pp. 4771–4787, 2019.
- [3] Y. Chen, S. Dong, T. Li, Y. Wang, and H. Zhou, "Dynamic multi-key FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5239–5249, 2021.
- [4] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
- [5] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, 2021.
- [6] S. B. Almin and M. Chatterjee, "A novel approach to detect android malware," *Procedia Computer Science*, vol. 45, pp. 407–417, 2015.
- [7] Li Jin, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, "Significant permission identification for machine-learning-based android malware detection," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 3216–3225, 2018.
- [8] L. Chen, S. Hou, Y. Ye, and L. Chen, "An adversarial machine learning model against android malware evasion attacks," in *Proceedings of the Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data*, pp. 43–55, Springer, Beijing, China, July 2017.
- [9] J. Zhang, Q. Zheng, H. Yin, Ou Lu, and Y. Hu, "IRMD: malware variant detection using opcode image recognition," in *Proceedings of the 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 1175–1180, IEEE, Wuhan, China, December 2016.
- [10] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the 8th international symposium on visualization for cyber security*, pp. 1–7, Pittsburgh, PA, USA, July 2011.
- [11] Y.-D. Lin, Y.-C. Lai, C.-H. Chen, and H.-C. Tsai, "Identifying android malicious repackaged applications by thread-grained system call sequences," *Computers & Security*, vol. 39, pp. 340–350, 2013.
- [12] M. Alfonso, I. Martin, A. Guzman, and J. A. Hernandez, "Android malware detection from Google Play meta-data: selection of important features," in *Proceedings of the 2015 IEEE Conference on Communications and Network Security (CNS)*, pp. 701–702, IEEE, Florence, Italy, September 2015.
- [13] D. Bryan and S. Mishra, "Power based malicious code detection techniques for smartphones," in *Proceedings of the 2013 12th IEEE international conference on trust, security and privacy in computing and communications*, pp. 142–149, IEEE, Melbourne, VIC, Australia, July 2013.
- [14] K. ElMouatez Billah, D. Mourad, D. Abdelouahid, and M. Djedjiga, "MalDozer: automatic framework for android malware detection using deep learning," *Digital Investigation*, vol. 24, pp. S48–S59, 2018.
- [15] C. Gerardo, D. Lorenzo Andrea, M. Eric, M. Francesco, and V. Corrado Aaron, "Effectiveness of opcode ngrams for detection of multi family android malware," in *Proceedings of the 2015 10th International Conference on Availability, Reliability and Security*, pp. 333–340, IEEE, Toulouse, France, August 2015.
- [16] D. Li, L. Zhao, Q. Cheng, N. Liu, and W. Shi, "Opcode sequence analysis of Android malware by a convolutional

- neural network,” *Concurrency and Computation: Practice and Experience*, vol. 32, Article ID e5308, 2020.
- [17] B. Zhang, W. Xiao, X. Xiao, A. K. Sangaiah, W. Zhang, and J. Zhang, “Ransomware classification using patch-based CNN and self-attention network on embedded N-grams of opcodes,” *Future Generation Computer Systems*, vol. 110, pp. 708–720, 2020.
- [18] P. Naser and X. Zhu, “Machine learning for android malware detection using permission and api calls,” in *Proceedings of the 2013 IEEE 25th international conference on tools with artificial intelligence*, pp. 300–305, IEEE, Herndon, VA, USA, May 2013.
- [19] A. Monte, M. Favero, G. Abed, J. Barroso, and P. Lício, “Identifying Android malware using dynamically obtained features,” *Journal of Computer Virology and Hacking Techniques*, vol. 11, pp. 9–17, 2015.
- [20] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, and K. Rieck, “Drebin: effective and explainable detection of android malware in your pocket,” *Ndss*, vol. 14, pp. 23–26, 2014.
- [21] H. Zhang, J. Qin, B. Zhang, H. Yan, J. Guo, and F. Gao, “A multi-class detection system for android malicious apps based on color image features,” in *Proceedings of the International Conference on Security and Privacy in New Computing Environments*, pp. 186–206, Springer, Tianjin, China, April 2020.
- [22] W. Han, J. Xue, Y. Wang, L. Huang, Z. Kong, and L. Mao, “MalDAE: detecting and explaining malware based on correlation and fusion of static and dynamic characteristics,” *Computers & Security*, vol. 83, pp. 208–233, 2019.
- [23] R. Feng, S. Chen, X. Xie, G. Meng, S.-W. Lin, and Y. Liu, “A performance-sensitive malware detection system using deep learning on mobile devices,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1563–1578, 2020.
- [24] A. Saba, A. Shah Munam, A. Wahid, A. Mehmood, H. Song, and H. Yu, “Samadroid: a novel 3-level hybrid malware detection model for android operating system,” *IEEE Access*, vol. 6, pp. 4321–4339, 2018.
- [25] S.-T. Guillermo, D. Santanu Kumar, A. Mansour, K. Johannes, G. Giorgio, and C. Lorenzo, “Droidsieve: fast and accurate classification of obfuscated android malware,” in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, pp. 309–320, Texas, America, March 2017.
- [26] B. Holland, D. Tom, K. Suresh, J. Mathews, and R. Nikhil, “Security toolbox for detecting novel and sophisticated android malware,” vol. 2, pp. 733–736, in *Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 2, pp. 733–736, IEEE, Florence, Italy, May 2015.
- [27] D. Quan, L. Zhai, Y. Fan, and P. Wang, “Detection of android malicious apps based on the sensitive behaviors,” in *Proceedings of the 2014 IEEE 13th international conference on trust, security and privacy in computing and communications*, pp. 877–883, IEEE, Beijing, China, September 2014.
- [28] D.-J. Wu, C.-H. Mao, T.-E. Wei, and H.-M. Lee, K.-P. W. Droidmat, Android malware detection through manifest and api calls tracing,” in *Proceedings of the 2012 Seventh Asia Joint Conference on Information Security*, pp. 62–69, IEEE, Tokyo, Japan, August 2012.
- [29] A. Ananya, A. Aswathy, T. R. Amal, P. G. Swathy, P. Vinod, and S. Mohammad, “SysDroid: a dynamic ML-based android malware analyzer using system call traces,” *Cluster Computing*, vol. 23, pp. 1–20, 2020.
- [30] Y.-l. Tang, X.-y. Li, Z.-q. Zhao, and Y.-F. Li, “Detection method for android payment cracked application,” *Journal of Beijing University of Posts and Telecommunications*, vol. 44, no. 4, p. 95, 2021.

Research Article

BCST-APTS: Blockchain and CP-ABE Empowered Data Supervision, Sharing, and Privacy Protection Scheme for Secure and Trusted Agricultural Product Traceability System

Guofeng Zhang ¹, Xiao Chen ², Bin Feng ¹, Xuchao Guo ³, Xia Hao ⁴,
Henggang Ren ¹, Chunyan Dong ¹ and Yanan Zhang ⁵

¹School of Information Science and Technology, Taishan University, Taian, Shandong 271000, China

²School of Economics and Management, Taishan University, Taian, Shandong 271000, China

³College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China

⁴College of Information Science and Engineering, Shandong Agricultural University, Taian, Shandong 271000, China

⁵School of Information Science and Engineering, University of Jinan, Jinan, Shandong 250022, China

Correspondence should be addressed to Guofeng Zhang; zhangguofeng@tsu.edu.cn and Bin Feng; binfeng@tsu.edu.cn

Received 23 October 2021; Revised 1 December 2021; Accepted 24 December 2021; Published 15 January 2022

Academic Editor: Yuling Chen

Copyright © 2022 Guofeng Zhang 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 provides new technologies and ideas for the construction of agricultural product traceability system (APTS). However, if data is stored, supervised, and distributed on a multiparty equal blockchain, it will face major security risks, such as data privacy leakage, unauthorized access, and trust issues. How to protect the privacy of shared data has become a key factor restricting the implementation of this technology. We propose a secure and trusted agricultural product traceability system (BCST-APTS), which is supported by blockchain and CP-ABE encryption technology. It can set access control policies through data attributes and encrypt data on the blockchain. This can not only ensure the confidentiality of the data stored in the blockchain, but also set flexible access control policies for the data. In addition, a whole-chain attribute management infrastructure has been constructed, which can provide personalized attribute encryption services. Furthermore, a reencryption scheme based on ciphertext-policy attribute encryption (RE-CP-ABE) is proposed, which can meet the needs of efficient supervision and sharing of ciphertext data. Finally, the system architecture of the BCST-APTS is designed to successfully solve the problems of mutual trust, privacy protection, fine-grained, and personalized access control between all parties.

1. Introduction

Food for the people: food safety is the first and most important. From a global perspective, food safety incidents are typical public health emergencies. In order to solve those, countries around the world have successively studied and established a variety of APTSs relying on the agricultural product supply chain, mainly using the centralized technical architecture to realize the shared storage of traceability data. However, frequent privacy leaks in the data center and frequent food safety incidents have led consumers to lose trust in the traceability system. At the same time,

considering many factors such as data ownership, data leakage, and their own commercial interests, agricultural production enterprises or organizations with a large amount of data are extremely cautious about opening their own internal data, especially core data. When food safety incidents break out, data are not available, tampered, or maliciously forged from time to time, resulting in the problems of less data and low reliability of agricultural product traceability system (APTS).

The main reasons for the above problems are as follows. Firstly, the data privacy of the participants in the supply chain is not effectively protected, resulting in the difficulty of

establishing a trust relationship between the participants. Secondly, regulators lack safe and effective regulatory technical means to effectively supervise the complete supply chain data. Finally, consumers no longer trust existing traceability systems and technologies. It can be seen that the contradiction between data privacy protection and efficient sharing of APTS is becoming increasingly prominent, and the problem of data security is still the difficulty and pain point restricting the safe sharing and supervision of agricultural products traceability data. The reason lies in the imperfect data privacy protection and access control technology of the traceability system.

The decentralization, nontampering, and traceability of blockchain technology provide new technologies and ideas for the construction of APTS. Based on blockchain and CP-ABE encryption technology, this paper constructs a secure and trusted agricultural product traceability system (BCST-APTS), which can meet the whole supply chain data supervision, fine-grained authorized access control, and secure and trusted data sharing.

The main contributions of this paper are as follows:

- (1) With the help of cryptographic algorithms, the data stored on the blockchain can be encrypted to ensure data privacy and completely solve the trust problem between consumers and system participants.
- (2) Based on CP-ABE encryption technology, it provides new technical means to solve the problems of privacy protection, fine-grained access control, and data supervision of agricultural product supply chain data.
- (3) The proposed attribute management infrastructure scheme can more efficiently and flexibly meet the personalized privacy protection needs of supply chain participants.
- (4) A RE-CP-ABE scheme is proposed and elaborated in detail, which can quickly and accurately determine data access rights. More importantly, it can meet the data supervision requirements of the supervisory organization for the entire supply chain.

2. Related Works

2.1. Agricultural Product Traceability System Based on Blockchain. Graves et al. [1] believe that the three processes of production, transportation, and sales are the core, and integrated production, information sharing, and production operations are an important research direction of the supply chain. Cachon and Fisher [2] believe that information sharing can effectively improve the operational efficiency of the supply chain. Boehlje et al. [3] believe that building a traceability system for agricultural products can effectively reduce the cost of food supervision and improve the quality of products. Gao et al. [4] believe that the establishment of trust mechanisms and information sharing mechanisms should be accelerated between all entities in the supply chain, and an information service platform should be built to realize corporate information sharing, so as to reduce the

overall operating costs of the supply chain and improve the operating efficiency and economic benefits of the supply chain. However, the existing data sharing and traceability system, which mainly adopts centralized technology architecture construction, can no longer be accepted by consumers. More precisely, the actual value of the traceability system is gradually being weakened.

The system architecture based on blockchain technology has the characteristics of decentralization, nontampering, traceability, etc., which can not only meet the traceability requirements of the entire process of the agricultural product supply chain, but also realize the distributed shared storage of agricultural product entire process data. Agricultural blockchain technology can make traceable information fairer, just, transparent, lightweight, and efficient to reach consensus [5]. However, the consensus mechanism is a key technology to achieve consensus between organizations and nodes on the chain, and its vulnerability may damage the entire blockchain system [6, 7]. Liu et al. [8] designed an anticounterfeiting traceability system based on blockchain technology that combines public and private chains to ensure the authenticity and reliability of the traceability information obtained and solve the problem of difficult supervision of traditional traceability systems. Feng [9] established an agricultural food supply chain traceability system based on RFID and blockchain technology. The system covers all links of the agricultural product supply chain, including the whole process of data acquisition and information management, and realizes the quality and safety monitoring, tracking, and traceability management of agricultural products “from farm to table.” Yang et al. [10] designed a “database+blockchain” agricultural product traceability information storage model and query method based on hyperledger fabric, and the encrypted hash value of traceability data is stored on the blockchain.

The above research successfully focused on the system architecture design and function realization, realized the distributed storage of agricultural product data, and ensured the data integrity. However, the existing research lacks in-depth research on data confidentiality, secure storage, access control, etc., cannot protect the data and privacy of entities in the traceability system, and is difficult to apply in practice.

2.2. Privacy Protection and Access Control of Blockchain. According to the degree of openness of the blockchain system, it can be divided into Public Blockchain, Private Blockchain, and Consortium Blockchain. According to whether the access of the organization node needs to be licensed, it can be divided into Public Blockchain and Permissioned Blockchain. Obviously, nonpublic Blockchains such as Consortium Blockchain and Private Blockchain are called Permissioned Blockchain [11]. Since the Permissioned Blockchain is a type of blockchain that each node needs to be licensed by the regulatory agency or authoritative organization, after verifying the identity, it is assigned specific system permissions to carry out specific businesses. Compared with the Public Blockchain, the Permissioned Blockchain is more suitable for application

scenarios that require supervision, cross-organization sharing, and multiparty business collaboration.

For any industry, users are unwilling to share their personal information and confidential data with competitors [12], such as source location privacy [13]. The design of the agricultural product supply chain scheme based on the blockchain should ensure the security and credibility of data encryption storage, transaction records can be traced, inquired, and appealable, and private data belongs to each participant [14]. In order to solve the data security problems faced by the traditional APTS, it is necessary to protect the privacy of participants in the whole agricultural product industry chain, based on safe and reliable data sharing, improve the enthusiasm of agricultural industrial organizations to participate in the construction and application of traceability system, and strengthen the effective supervision of regulatory agencies, enhancing consumers' confidence and satisfaction with the traceability results. Hyperledger blockchain is committed to providing new solutions for data security and privacy protection [15, 16]. For example, Hyperledger fabric has been used in the pharmaceutical traceability system [12]. The APTS fully meets the above characteristics, which is also the key application field of blockchain in agriculture.

Access control is the core key technology for data privacy protection. Through access permissions, data can only be accessed by the owner and authorized legal users. At present, the Permissioned Blockchain mainly adopts technologies such as organization (user) identity authentication, privacy channel, main/subchain data isolation [17], multi-subchain model [18, 19], endorsement strategy, transaction encryption, smart contract encryption, and privacy data set to realize access control of block data. Organizational identity authentication solves the access control problem at the blockchain network level and prevents unauthorized users from entering the blockchain network; the privacy channel realizes the logical isolation between the organizations inside and outside the channel and achieves access control at the channel level, but it has different circumstances, creating a separate privacy channel that will incur additional management overhead (such as maintaining chain code version, policy and Membership Service Provider (MSP)). Obviously, the main/subchain data isolation and multichain model also have the same kind of problems as mentioned above, and endorsement policy can realize the organizational level access control of smart contract writing, but there is a risk of privacy disclosure due to cross-channel unauthorized access, and transaction encryption and smart contract encryption mechanisms still remain at the channel level; privacy dataset can realize access control of privacy data without creating a new privacy channel, but it still stays at the organizational level.

None of the above technologies can achieve more fine-grained (such as organization-level/node-level) access control to meet the complex access requirements of the Permissioned blockchain across organizations [20], and other access control technologies are still needed. Fabric CA 1.4 version has adopted Attributes-based Access Control (ABAC), through the organization of identity attributes to

access control of smart contract (chain code) operations, but it still lacks flexibility to set attributes only from the perspective of organizational identity. At the same time, the confidentiality of shared data cannot be guaranteed. Wang et al. [21] proposed an Attribute-based Distributed Access Control Framework (ADAC) suitable for IoT blockchain. Based on ABAC and blockchain, Zhang et al. [22] use the access tree [23] to configure access policies to achieve fine-grained authorized access to IoT devices. ABE is also used for access control of data sharing under the blockchain. Alniamy and Taylor [24] proposed fine-grained access control of shared data under the distributed environment of the blockchain. Jemel and Serhrouchni [25] and Huang et al. [26] solved the problem of fine-grained access control faced by data protection in an open shared environment, but the attribute set is open to all nodes in the entire network, which can easily be stolen by malicious nodes to generate correct users Key. Wang et al. [20] used ABE to propose a data access control and sharing model to achieve fine-grained access control and secure sharing. With the increasing number of on-chain organizations, when cross-organization deployment increases information sharing between different organizations, ABAC implementation may become complicated and requires attribute management infrastructure [27].

However, the above-mentioned existing research only focuses on the design of fine-grained access control and does not provide an overall plan that includes attribute management infrastructure and effective supervision of encrypted data, which is not conducive to the unified supervision of encrypted data by supervision organizations.

2.3. Block Data Encryption and Flexible Sharing. Use blockchain distributed ledger and encryption technology to realize the privacy protection and safe sharing of agricultural global data, so as to ensure the stability of agricultural system operation and ensure the business flow (information flow), capital flow, and logistics data of the entire agricultural industry chain authenticity [5]. Data confidentiality is a prerequisite to ensure data security. Block (ledger) data security mainly encrypts transaction data through cryptographic algorithms. Symmetric encryption system can be used for blockchain data encryption [20, 28]. This system requires both encryption and decryption parties to share keys. The ciphertext data can be calculated using a multikey fully homomorphic encryption (MFHE) scheme. Chen et al. proposed a dynamic multikey FHE scheme based on the LWE assumption [29], which requires less "local" memory, and the ciphertext expansion process is distributed. With the increasingly complex business exchanges between organizations and the dynamic changes in the number of organizations, key distribution and management will become complicated and difficult to operate. At the same time, there will be key leakage and multiple encryption problems. If the entire blockchain uses the same cryptographic algorithm and key, it is meaningless for data protection in the blockchain. What is more dangerous is that once an organization or node is illegally compromised, the loss is

immeasurable. Obviously, symmetric cryptosystem is not the best choice for blockchain data encryption.

Relatively speaking, a public key cryptosystem based on Public-Key Infrastructure (PKI) is more suitable. At present, blockchains mostly use public key cryptosystems to encrypt data [20]. Although they have high security, they are limited to data sharing between the two, which cannot meet the data sharing of 1-to-N and multilevel access control [20]. In order to support more flexible public key generation, Sahai and Waters [30] proposed an Attribute-based Encryption (ABE) scheme, which uses a series of attribute sets instead of unique identifiers to identify identities. ABE is a fine-grained 1-to-N encryption scheme. Its advantages are as follows:

- (1) Encryption is only related to attributes, without paying attention to the number and identity of access members, which reduces the encryption overhead
- (2) Only the members that conform to the ciphertext attribute can be decrypted, so as to ensure the security of the data
- (3) The key is related to random numbers, and the keys of different members cannot be combined, which can resist collusion attacks [20]

Further research proposes Key-policy Attribute-based Encryption (KP-ABE) [31] and Ciphertext-policy Attribute-based Encryption (CP-ABE) [32]. KP-ABE embeds the policy into the encryption key and the attribute into the ciphertext. The key corresponds to an access structure and the ciphertext corresponds to a set of attributes. CP-ABE embeds the policy into the ciphertext and the attribute into the user key. The ciphertext corresponds to an access structure, and the key corresponds to a set of attributes. The common feature of the two is to bind data encryption and decryption with policy. The data can be decrypted only when the attributes in the attribute set can meet the access structure. While retaining the ciphertext control, fine-grained access control can be realized. KP-ABE scheme is close to static scenarios, such as paid video websites and log encryption management. In CP-ABE scheme, the data owner specifies the strategy of accessing ciphertext and associates the attribute set with the access resources. Data users can access ciphertext data according to their own attributes. This technology is suitable for access applications such as private data sharing, such as data encryption storage and fine-grained sharing in cloud computing environment.

In view of the above analysis, this paper uses CP-ABE scheme to encrypt the data stored in the APTS, which can not only protect the data privacy and security of the uplink organization, but also lay a foundation for flexible data sharing.

3. BCST-APTS: Secure and Trusted Agricultural Product Traceability System

3.1. System Logic Architecture. A secure and trusted agricultural product traceability system covers the entire process of production, processing, warehousing, logistics, and sales in the agricultural product supply chain. Participating

entities include farmers/producers, processors, warehouse operators, logistics providers, retailers, and consumers. The business of each participant is carried out under the effective supervision of the Regulatory authority. The regulatory authority is responsible for the identity authentication, authority management, data supervision, and traceability of agricultural product quality and safety events for each subject. The system logic architecture is shown in Figure 1.

The system realizes the whole process data collection of agricultural products “from farm to table,” that is, preproduction data, mid-production data, and postproduction data, including structured data and unstructured data. Structured data can be encrypted and stored directly on the blockchain, and unstructured data can be stored off blockchain, but its digital fingerprints must be stored on the blockchain to ensure the integrity and confidentiality of the data. Based on Permissioned blockchain and data encryption technology, the system has the following technical characteristics.

- (1) *No Tampering.* Ensure the authenticity, validity, and permanence of data stored on the chain.
- (2) *Distributed Storage.* Sharing by members of the whole blockchain avoids the technical risks of centralized architecture.
- (3) *Data Encryption and Flexible Access Control.* It can protect the privacy of the data publisher and solve the problem of separation of data ownership and control on the blockchain.
- (4) *Tracing Smart Contract.* When an agricultural product quality and safety incident occurs, the data of relevant participants can be automatically extracted and uploaded to the system, so as to prevent the relevant parties from tampering, deleting, or forging data when the incident occurs, so as to restore the truth of the incident and find the root cause of the problem.

In the above architecture, data encryption and flexible access control are the keys to ensuring that this system has the characteristics of security and credibility. It is also a typical difference between this work and other agricultural product traceability systems based on blockchain technology. In order to achieve the unification of the two, this paper focuses on the realization of the encryption and fine-grained access control of the data on the blockchain based on the CP-ABE scheme. The reencryption scheme based on ciphertext policy attribute encryption (RE-CP-ABE) is introduced in detail in Section 4.

3.2. System Deployment Network Architecture. As mentioned in Section 3.1, the BCST-APTS involves multiple participants in the agricultural product supply chain. At present, in order to achieve efficient management within the enterprise, each entity has built a relatively complete information system, but the business system of each entity has huge differences in business logic, technical architecture, and deployment plans. Therefore, so as to achieve various

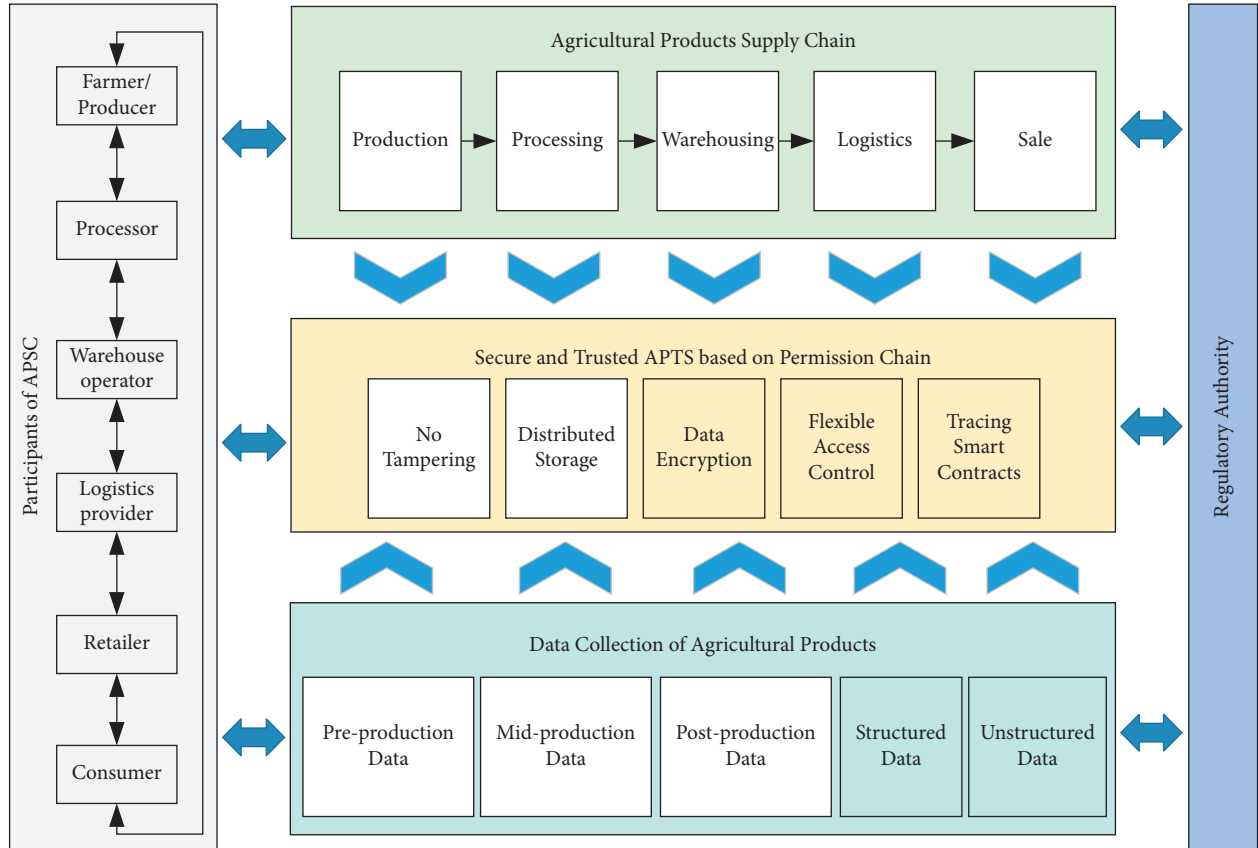


FIGURE 1: The system logic architecture of BCST-APTS.

business alliances, data sharing and business systems between subjects must solve the problems of multisource and heterogeneous internal business systems. The distributed characteristics of blockchain technology itself provide a new solution to the above problems. Figure 2 shows the deployment network architecture diagram of the system.

As shown in Figure 2, between the internal business system of each participant and the blockchain system, one or more blockchain nodes are built, and the internal business system and BCST-APTS are realized with the help of the client. For seamless connection of the blockchain, such as Organization A, Organization B, and Organization C as different participants in the agricultural product supply chain, there is also a regulatory organization responsible for supervision and operation of the entire blockchain system.

Note. The regulatory organization here is not a traditional centralized agency; it is just one of the ordinary members on the blockchain. When agricultural products need to be traded, the relevant data is packaged, and private data and trade secret data are encrypted using the CP-ABE encryption algorithm. The encrypted ciphertext is released and stored on the blockchain through the blockchain node, and data retrieval is only completed on the local blockchain node.

From the perspectives of part of the enterprise and the entire chain as a whole, the system architecture has obvious

advantages. First of all, from the perspective of the organization, not only can the stability of the internal business system be ensured, but also a secure and reliable blockchain system can be accessed. Secondly, from the overall perspective of the entire chain, all participants jointly maintain a set of ledger books to achieve cross-regional and cross-industry agricultural product traceability business collaboration and data sharing, so as to ensure the authenticity and credibility of agricultural product traceability.

4. Reencryption Scheme Based on Ciphertext-Policy Attribute Encryption (RE-CP-ABE)

4.1. CP-ABE Scheme Features. The data in the blockchain ledger is open to the whole nodes, which cannot guarantee the confidentiality of the data and is easy to be accessed illegally. This paper introduces CP-ABE encryption scheme to ensure the data confidentiality and authorized access control of the data sharers and realize the unity of data ownership and control on the blockchain. CP-ABE Encryption Scheme [32] consists of five basic algorithms, including setup, encrypt, keygen, decrypt, and delegate. Among them, $CT = \text{encrypt}(PK, m, t)$ is an encryption algorithm. The encryption algorithm encrypts a message m under the tree access structure T . The specific calculation formula is as follows:

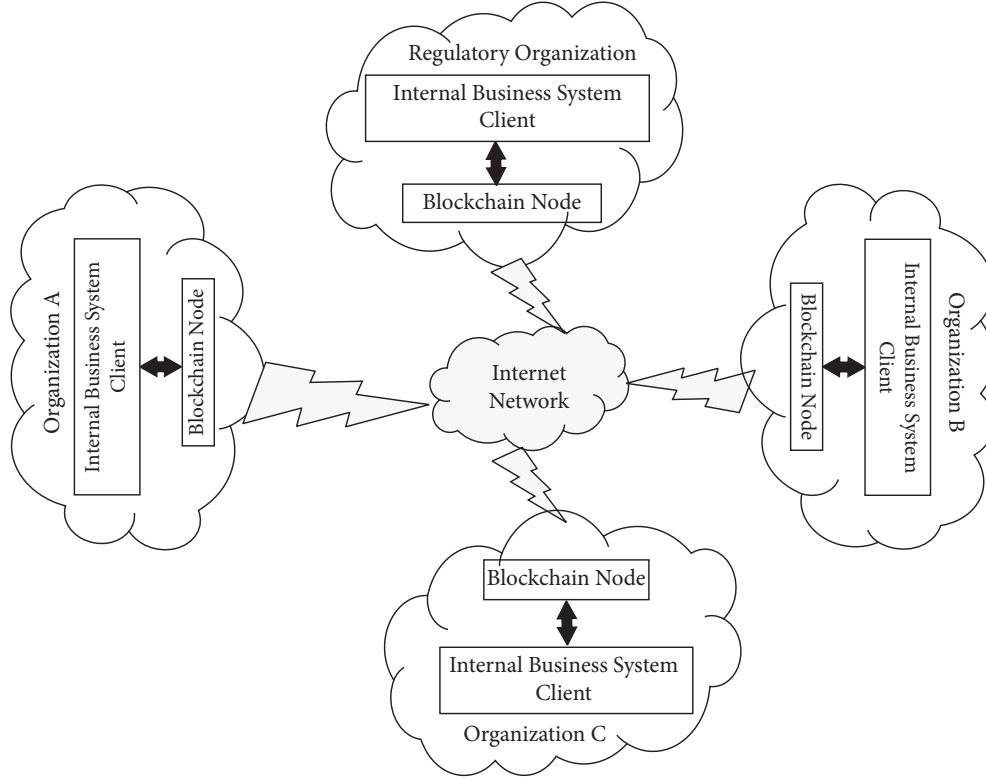


FIGURE 2: System deployment network architecture of BCST-APTS.

$$CT = (T, \check{C} = \text{Me}(g, g)^{as}, C = h^s, \forall y \in Y: C_y = g^{q_y^{(0)}}, C'_y = H(\text{att}(y))^{q_y^{(0)}}). \quad (1)$$

Here, the ciphertext CT is constructed by T , which is the tree access structure. The function $\text{att}(x)$ is defined only if x is a leaf node and denotes the attribute associated with the leaf node x in T .

The decryption function is $\text{DecryptNode}(CT, SK, x)$, defined as

$$\text{DecryptNode}(CT, SK, x) = \frac{e(D_i, C_x)}{e(D'_i, C'_x)} = \frac{e(g^r \cdot H(i)^{r_i}, h^{q_x^{(0)}})}{e(g^{r_i}, H(i)^{q_x^{(0)}})} = e(g, g)^{r q_x^{(0)}}. \quad (2)$$

Here, SK is a private, which is associated with a set S of attributes, and a node x from T .

Reference [32] explains the meaning of other parameters in detail, which will not be repeated here. However, from the above two formulas and parameters T and $\text{att}(x)$, it can be seen that, in CP-ABE algorithm, the attribute is extremely important for data encryption, decryption, and access control. It determines the flexibility of access control policy and who can decrypt ciphertext data. However, in order to meet the personalized encryption needs of each subject accessing the APTS, the system should support the needs of each subject to set personalized attributes, but it will lead to the increase of attribute synonymy or redundancy. At the same time, it is not conducive to the efficient supervision of encrypted data by regulators.

4.2. Access Control Tree. Figure 3 shows an access control tree model in Apple's traceability system. In order to show the principle, it only includes four parts: product type, brand, place of production, and logistics provider.

It can be seen from Figure 3 that leaf nodes represent an attribute of shared data, and non-leaf nodes are threshold nodes that support "AND" or "OR" logic operations. Data requesting organization must meet the minimum threshold value before they can decrypt the secret value of this node. For example, the threshold node "1/2" means that at least one of the two attributes can be decrypted, which is one of JD.com or SF Express. When the data requesting organization applies for access to encrypted data, only users who have the attributes in the access control tree and satisfy the logical relationship can access, so that the data can be encrypted once and shared N times.

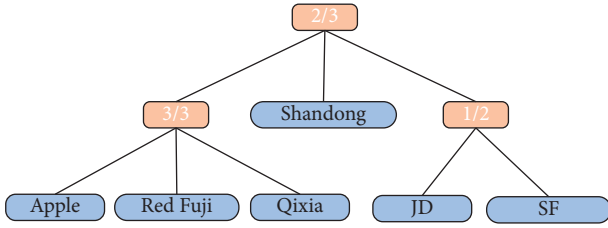


FIGURE 3: Access control tree model in Apple's traceability system.

4.3. Attribute Management Infrastructure. To solve the above-mentioned problems, this paper proposes that the authoritative organization or regulatory authority in the APTS builds a whole-chain standardized attribute management infrastructure to provide attribute management, access, and other services to all access organizations in the entire blockchain. The specific construction process of this attribute management infrastructure is shown in Figure 4.

The construction of the attribute management infrastructure includes the following steps:

- (1) Initialization phase: at this stage, the authoritative organization establishes the structure and storage mode of the attribute management infrastructure and establishes the user attribute set to standardize the management of all attributes of the whole chain. The structure of attribute management infrastructure can adopt key value, relational table, etc., and be stored in the form of file or database table. The user attribute set is used to store all attribute sets owned by the organization.
- (2) Assign public attributes to the access organization. When approving the access application of each organization, the authoritative organization assigns public attributes to the application organization according to its business, role, etc. The public attributes can be organization name, organization identity ID, system role, access time, and other different contents.
- (3) The access organization applies for private attributes. After accessing the permissioned blockchain system, each organization can apply to an authoritative organization to maintain its own private attributes based on its own business development. The authoritative organization decides whether to approve the application. After passing the application, the organization can be used for subsequent data encryption and decryption.
- (4) Establish a whole-chain attribute management infrastructure. The public and private attributes of each organization together constitute the entire blockchain of attribute management infrastructure.
- (5) Maintain the attributes of the entire blockchain. The authoritative organization dynamically maintains and manages the attributes of the entire blockchain and the attribute collection in the attribute management infrastructure according to the result of the attribute application.

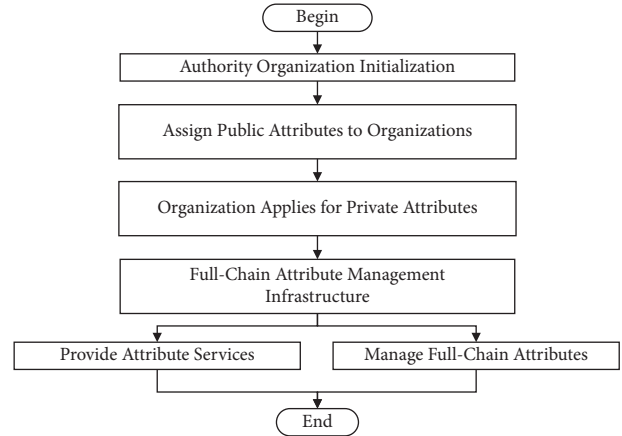


FIGURE 4: Flowchart of whole-chain attribute management infrastructure.

- (6) Provide attribute services. Authoritative organizations provide external attributes services such as query, modification, and deletion according to the attribute management infrastructure and the attribute collection of the organization. For example, the data issuer retrieves the attributes used for data encryption, and when the authoritative organization works in place of the CA, it can generate encryption keys based on the data issuer.

The above attribute management infrastructure construction method manages the attributes of the entire blockchain through the attribute dictionary, which can not only meet the personalized attribute requirements of different access organizations, but also convert redundant attributes and synonymous attributes into standardized and standard attributes. A flexible and efficient solution is proposed for the difficult problem of attribute management in attribute-based encryption schemes.

4.4. RE-CP-ABE Encryption Scheme. The CP-ABE encryption scheme can configure flexible and personalized encryption and access control policies with the help of attributes, but it also poses challenges for the entire blockchain of sharing and supervision of encrypted data. When the data issuer releases the encrypted data to the traceability system, if the original access control policy remains unchanged, it can be unified to the entire blockchain of standardized data encryption, so that the data owner, data requester, and data supervisors can quickly access data, which will greatly improve the management efficiency of the system. For this reason, RE-CP-ABE is proposed in this paper.

RE-CP-ABE scheme consists of six core algorithms: Setup, Encrypt, UpBlockChain, ReEncrypt, AccessKeyGen, and Decrypt. All variable symbols used in the specific algorithm are shown in Table 1.

$$\text{Setup}() \longrightarrow \text{PK, MK.} \quad (3)$$

TABLE 1: Symbolic variable.

Variable name	Meaning
Setup	System initialization algorithm
PK	Public parameters
MK	Master key
Encrypt	Personalized encryption algorithm
M	Plaintext message
T	Personalized access control tree
CT	Personalized ciphertext
UpBlockChain	Block publishing algorithm
ReEncrypt	Attribute re-encryption algorithm
CT'	Standardized ciphertext
T'	Standardized access control tree
AccessKeyGen	Access control and key generation algorithm
S	Personalized attribute set selected by the data request user
SK	Decryption private key
Decrypt	Data decryption algorithm

System initialization algorithm: it has no input parameters, output public parameters PK, and master key MK.

$$\text{Encrypt}(\text{PK}, M, T) \longrightarrow \text{CT}. \quad (4)$$

Personalized encryption algorithm: according to the personalized access control tree T , it is constructed by users according to their own personalized needs, flexibly selected attribute set U_p and logical relations, and personalized encryption is performed on the plaintext message M to obtain a personalized ciphertext CT.

$$\text{UpBlockChain}(\text{CT}, T). \quad (5)$$

Block publishing algorithm. Publish the encrypted personalized ciphertext CT and the corresponding access control tree T to the authoritative organization node or block generation node of the blockchain system, such as the Orderer node of Fabric.

$$\text{ReEncrypt}(\text{PK}, \text{CT}, T) \longrightarrow \text{CT}', T'. \quad (6)$$

Attribute reencryption algorithm: this algorithm is executed by an authoritative organization node and uses the attribute service provided by the attribute management infrastructure to reencrypt the received personalized ciphertext CT into a standardized ciphertext CT' . At the same time, the personalized access control tree T is converted into a standardized access control tree T' .

$$\text{AccessKeyGen}(T', S) \longrightarrow \text{SK}. \quad (7)$$

Access control and key generation algorithm: the algorithm is executed by the authoritative organization node and uses the attribute service provided by the attribute management infrastructure to determine whether the personalized attribute set S selected by the data request user meets the standardized access control tree T' . If both the attributes and the logical relationship meet the requirements, the user's data decryption private key SK is generated. Otherwise, there is no access control authority, and the decryption private key SK cannot be obtained.

$$\text{Decrypt}(\text{PK}, \text{CT}', \text{SK}) \longrightarrow M. \quad (8)$$

Data decryption algorithm: according to the system public parameter PK and the decryption private key SK, the standardized data ciphertext CT' is decrypted into plaintext message M .

This algorithm is an improvement of the CP-ABE [32] scheme and retains the technical advantages of the original algorithm that flexibly set access control policies and data encryption according to attributes. At the same time, with the help of a standardized access control tree T' , it is possible to realize the standardization of personalized data encryption and access control, so that access rights can be quickly determined, and the effective supervision of encrypted data by data supervision organizations and authoritative third parties can be ensured.

5. BCST-APTS Based on Fabric and RE-CP-ABE

5.1. BCST-APTS Scheme. As one of the typical representatives of the Permissioned Blockchain, the fabric has been widely studied and applied in various fields. It realizes the technical positioning of business collaboration for alliance members, which determines that it can be successfully applied to the traceability system of agricultural products. This paper designs a secure and trusted agricultural product traceability system scheme based on Fabric and RE-CP-ABE, as shown in Figure 5.

This system scheme consists of data publisher (Organization 1), data requester (Organization N), and authoritative organizations, and the authoritative organization is responsible for the operation and maintenance management of the CA node and Orderer node of the system. Each connected organization manages its own Peer node and saves a copy of the blockchain ledger with the entire chain data.

5.2. Business Process. The specific business process of the scheme is as follows:

- (1) Data is encrypted and published on the blockchain. The data publisher (organization 1) uses the

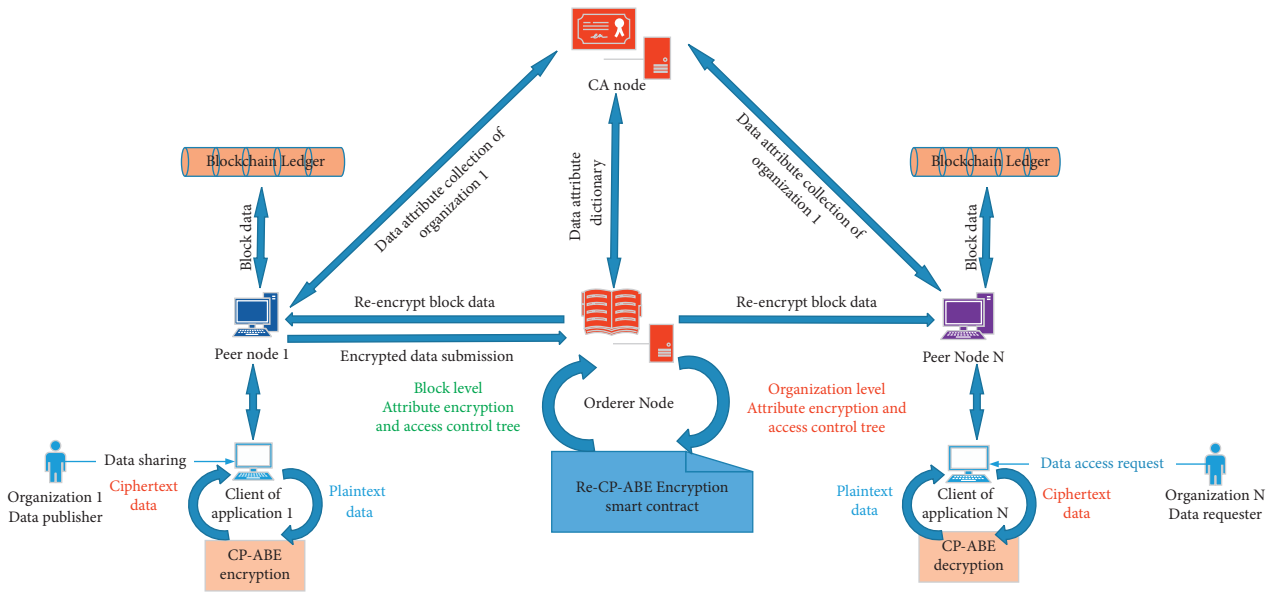


FIGURE 5: BCST-APTS scheme based on fabric and RE-CP-ABE.

application 1 client to interact with Peer node 1, which is a node managed by itself, for blockchain interaction. The client is responsible for selecting attributes for encryption and constructing an access tree. After attribute encryption, the ciphertext data is submitted to the Peer node for publishing on the blockchain.

- (2) Reencryption by authoritative organizations: the RE-CP-ABE encrypted smart contract is deployed in the Orderer node, and the Orderer node uses it to reconstruct the access tree of shared data, that is, converting it to the canonical access tree under the attribute dictionary, and reencrypting the data. As a result, the ciphertext data (organization level) encrypted by different organization attributes is converted to the ciphertext data (block level) under the specification attributes of the whole-chain attribute dictionary.
- (3) Store standard ciphertext data blocks: the Orderer node is responsible for sorting the received transactions, generating blocks of reencrypted data, broadcasting them to each Peer node 1 and Peer node N on the permissioned blockchain, and writing them into the blockchain ledger.
- (4) The data requester decrypts: the data requester (organization N) sends the data request and the attributes it owns to the Peer node N through the client of the application N. Then, the Peer node N automatically executes the smart contract constructed by the canonical access tree to check whether the attributes and access control of data access are met condition. If it is determined to be satisfied, it returns to the client a request response including encrypted data and a standardized access control tree. The application N client decrypts the returned encrypted data to obtain the plaintext data.

5.3. Security Analysis. In the proposed scheme, the CP-ABE encryption algorithm is used to protect data privacy and access control, the blockchain technology is used to ensure the distributed storage of data, and the RE-CP-ABE scheme is designed to ensure efficient data supervision of encrypted data to ensure the security and efficiency of the design scheme.

- (1) Data confidentiality: this solution uses CP-ABE to encrypt the data on the blockchain and stores the ciphertext data on the blockchain. Although all nodes of the blockchain can obtain the data, the data content cannot be obtained when the attributes and access control tree requirements are not met. Therefore, data privacy and security are protected.
- (2) Data integrity: this solution stores the traceability data of agricultural products on the blockchain. With the help of the chain storage structure of the blockchain ledger, it can effectively prevent a single node from tampering with the data and ensure the integrity of the traceability data.
- (3) Data availability: all nodes participating in the traceability system can have a copy of the complete ledger. Therefore, when the service of a single node or multiple nodes is abnormal or interrupted, the entire system can still operate normally, which can effectively guarantee the availability of the system and data.
- (4) Binding security of data ownership and control rights: taking full advantage of the fine-grained access control technology of the CP-ABE algorithm, it solves the problem of the data owner's control of distributed storage data on the blockchain and, at the same time, realizes one-time encryption for data release and multiple authorizations for data access, thereby improving the security and flexible access flexibility of the data on the blockchain.

- (5) Reencryption security: the RE-CP-ABE scheme designed in this paper is implemented by the Orderer node of an authoritative organization, which can effectively identify malicious attribute operations such as forgery and impersonation by participants, thereby further ensuring the security of reencrypted data.

6. Conclusion and Prospect

In this paper, blockchain technology and CP-ABE algorithm are successfully integrated and applied to a secure and trusted agricultural product traceability system (BCST-APTS). Furthermore, an attribute management infrastructure is designed, which can regulate and efficiently manage the attributes of the entire blockchain. Based on this and CP-ABE algorithm, a RE-CP-ABE scheme is proposed, which can convert personalized encryption to standardized encryption, thereby ensuring the efficient sharing and supervision of data stored in the Permissioned Blockchain. Finally, this paper designs a BCST-APTS scheme based on Fabric and RE-CP-ABE. The above research work provides new solutions and ideas for solving the problems of data fraud, untrustworthy traceability results, and privacy leakage in the APTS. This work currently only designs the model of the system from the perspective of technology, architecture, and principles. The follow-up will focus on an in-depth research on the security of smart contracts, the efficiency of attribute management infrastructure, the flexibility and efficiency of the RE-CP-ABE solution, and the final construction a complete and usable APTS serving the development of agricultural product traceability technology.

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 did not have any conflicts of interest.

Acknowledgments

This work was supported by the Project of Shandong Provincial Natural Science Foundation under Grant no. ZR2021QF056, National Natural Science Foundation of China under Grant nos. 62071320 and 61771090, Tai'an Science and Technology Innovation Development Project under Grant no. 2020NS080, and Shandong Federation of Social Sciences under Grant no. 2021-YYGL-32.

References

- [1] S. C. Graves, D. B. Kletter, and W. B. Hetzel, "A dynamic model for requirements planning with application to supply chain optimization," *Operations Research*, vol. 46, pp. 35–49, 1998.
- [2] G. P. Cachon and M. Fisher, "Supply chain inventory management and the value of shared information," *Management Science*, vol. 46, no. 8, pp. 1032–1048, 2000.
- [3] M. Boehlje, J. Akridge, and D. Downey, "Restructuring agribusiness for the 21st century," *Agribusiness*, vol. 11, no. 6, pp. 493–500, 1995.
- [4] J. Gao, J. Teng, L. Hou, and X. Liu, "Pricing strategy of closed-loop supply chain considering competition under uncertain demand," *Journal of Systems Engineering*, vol. 32, pp. 78–88, 2017, in Chinese.
- [5] W. Gao, G. Zhang, G. Zhang et al., "Original innovation of key technologies leading healthy development of smart agricultural," *Smart Agriculture*, vol. 1, no. 1, pp. 8–19, 2019, in Chinese.
- [6] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
- [7] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, pp. 1–22, 2021.
- [8] J. Liu, T. Yang, and W. Wang, "Traceability system using public and private blockchain," *Journal of Cyber Security*, vol. 3, no. 3, pp. 17–29, 2018, in Chinese.
- [9] T. Feng, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," in *Proceedings of the 2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, pp. 1–6, Kunming, China, June 2016.
- [10] X. Yang, M. Wang, D. Xu, N. Luo, and C. Sun, "Data storage and query method of agricultural products traceability information based on blockchain," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 35, no. 22, pp. 323–330, 2019, (in Chinese).
- [11] J. Zhu, Q. Ding, and S. Gao, "Distributed Framework of SWIFT system based on permissioned blockchain," *Journal of Software*, vol. 30, no. 6, pp. 1594–1613, 2019, (in Chinese).
- [12] M. Uddin, "Blockchain Meddler: hyperledger fabric enabled drug traceability system for counterfeit drugs in pharmaceutical industry," *International Journal of Pharmaceutics*, vol. 597, Article ID 120235, 2021.
- [13] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, 2021.
- [14] L. Yu, G. Zhang, J. Jia, and W. Gao, "Modern agricultural product supply chain based on block chain technology," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 48, pp. 387–393, 2017.
- [15] S. Namasudra, G. C. Deka, P. Johri, M. Hosseinpour, and A. H. Gandomi, "The revolution of blockchain: state-of-the-art and research challenges," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1497–1515, 2020.
- [16] A. A. Khan, M. Uddin, A. Shaikh, A. A. Laghari, and A. E. Rajput, "MF-ledger: blockchain hyperledger sawtooth-enabled novel and secure multimedia chain of custody forensic investigation architecture," *IEEE Access*, vol. 99, 2021.
- [17] X. Min, Q. Li, J. Kong, and D. Zhang, "Permissioned blockchain dynamic consensus mechanism based multi-centers," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 41, no. 5, pp. 1005–1020, 2018, (in Chinese).
- [18] Q. Ding, J. Zhu, J. Zhang et al., "Traceability permissioned chain consensus mechanism based on double-layer architecture," *Journal of Network and Information Security*, vol. 5, no. 2, pp. 1–12, 2019, (in Chinese).
- [19] W. Tsai, R. Blower, Y. Zhu, and L. Yu, "A system view of financial blockchains," in *Proceedings of the 2016 IEEE*

- Symposium on Service-Oriented System Engineering (SOSE)*, April 2016.
- [20] X. Wang, X. Jiang, and Y. Li, "Model for data access control and sharing based on blockchain," *Ruan Jian Xue Bao/Journal of Software*, vol. 30, no. 6, pp. 1661–1669, 2019, (in Chinese).
 - [21] P. Wang, Y. Yue, W. Sun, and J. Liu, "An attribute-based distributed access control for blockchain-enabled IoT," *Networking and Communications (WiMob)*, in *Proceedings of the 2019 International Conference on Wireless and Mobile Computing*, pp. 1–6, Barcelona, Spain, October 2019.
 - [22] Y. Zhang, B. Li, B. Liu, J. Wu, Y. Wang, and X. Yang, "An attribute-based collaborative access control scheme using blockchain for IoT devices," *Electronics*, vol. 9, no. 2, p. 285, 2020.
 - [23] A. Castiglione, A. Santis, B. Masucci et al., "Hierarchical and shared access control," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 850–865, 2016.
 - [24] A. Alniamy and B. D. Taylor, "Attribute-based access control of data sharing based on hyperledger blockchain," in *Proceedings of the ICBCT'20: 2020 The 2nd International Conference on Blockchain Technology*, pp. 135–139, Hilo, HI, USA, March 2020.
 - [25] M. Jemel and A. Serhrouchni, "Decentralized access control mechanism with temporal dimension based on blockchain," in *Proceedings of the IEEE International Conference on E-business Engineering IEEE*, pp. 177–182, Shanghai, China, November 2017.
 - [26] S. Huang, W. Chen, and B. Fan, "Data security sharing method based on CP-ABE and blockchain," *Computer Systems & Applications*, vol. 28, no. 11, pp. 79–86, 2019.
 - [27] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, "Attribute-based access control," *Computer*, vol. 48, no. 2, pp. 85–88, 2015.
 - [28] Z. Huang, "The application of blockchain in edge computing and IoT," *Cyberspace Security*, vol. 9, no. 8, pp. 25–30, 2018.
 - [29] Y. Chen, S. Dong, T. Li, Y. Wang, and H. Zhou, "Dynamic multi-key FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5239–5249, 2021.
 - [30] A. Sahai and B. Waters, "Fuzzy identity-based encryption," *Lecture Notes in Computer Science*, in *Proceedings of the International Conference on Theory & Applications of Cryptographic Techniques*, pp. 457–473, Aarhus, Denmark, May 2005.
 - [31] 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*, pp. 89–98, Alexandria Virginia, USA, October 2006.
 - [32] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security & Privacy*, pp. 321–334, IEEE, Berkeley, CA, USA, May 2007.

Research Article

Attribute Set-Based Boolean Keyword Search over Encrypted Personal Health Records

Yu Lin ¹, Lingling Xu ¹, Wanhua Li ¹ and Zhiwei Sun ²

¹School of Computer Science and Engineering, South China University of Technology, Guangzhou, China

²School of Artificial Intelligence, Shenzhen Polytechnic, Shenzhen, China

Correspondence should be addressed to Lingling Xu; csslxu@scut.edu.cn

Received 24 September 2021; Accepted 11 November 2021; Published 23 December 2021

Academic Editor: Xin-Yi Huang

Copyright © 2021 Yu Lin 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.

A personal health record (PHR) is an electronic application which enables patients to collect and share their health information. With the development of cloud computing, many PHR services have been outsourced to cloud servers. Cloud computing makes it easier for patients to manage their personal health records and makes it easier for doctors and researchers to share and access this information. However, due to the high sensitivity of PHR, a series of security protections are needed to protect them, such as encryption and access control. In this article, we propose an attribute set-based Boolean keyword search scheme, which can realize fine-grained access control and Boolean keyword search over encrypted PHR. Compared with the existing attribute-based searchable encryption, our solution can not only improve the flexibility in specifying access policies but also perform Boolean keyword search, which can meet the needs of large-scale PHR users. Furthermore, we simulate our scheme, and the experimental results show that our scheme is practical for PHR systems in cloud computing.

1. Introduction

In recent years, with the rapid development of cloud computing, it has been successfully deployed in a wide variety of real-world applications, such as the medical industry. A personal health record (PHR) is an electronic application through which patients can maintain and manage their health information in a private, secure, and confidential environment. The intention of PHR is to provide a complete and accurate summary of an individual's medical history which is accessible online. PHR benefits in many aspects, including strengthening the connection between doctors, patients, medical researchers, and hospitals. Doctors and medical researchers can obtain patient's information more conveniently by using PHR. At the same time, patients can better control their personal health information (PHI) since only people with necessary electronic credentials can get access to their PHR. However, PHR is under security threat such as information leakage, lack of access control, and untrusted cloud servers. A lot of security issues have happened in recent years, which have brought huge economic losses. Thus, it is very important to develop a method

for protecting the security and privacy of PHR. The most common method is to encrypt the PHR before uploading them to the server. However, performing searches on encrypted PHR would be a challenge.

Searchable encryption (SE) is a well-studied method that can deal with this issue. In SE, firstly, data owners encrypt their data and upload them to the cloud server. Then, each legitimate data user can generate a trapdoor using his secret key and interested keywords where the trapdoor enables data user to search over encrypted PHR. However, due to the high sensitivity, it is desired that each user can only query the authorized PHR. That is, fine-grained access control is needed for PHR. Thus, attribute-based keyword search has been implemented by many researchers [1–12]. In these schemes, data owners can encrypt their data with predefined access control policies. A trusted authority is responsible for managing all data users and distributing secret keys to them, where the secret key is generated according to the user's attributes. Thus, when a data user searches over encrypted PHR stored in the server with his interested keywords, the server can judge whether his attributes satisfy the access policy.

However, there are some limitations in existing schemes. Firstly, in the schemes [7–10], each user’s attributes are organized in a single set, which cannot support the compounded attributes. For example, consider a user who is a “Researcher” working in “College A” and serves as both “Director” of “Department of Medicine” and “Professor” of “Department of Chemistry,” and the above attributes are both valid and are likely to be used to describe him. A possible method is expressing above attributes as strings, like “Researcher_College A_Director_ Department Of Medicine_Professor_ Department Of Chemistry”. Unfortunately, it becomes challenging in satisfying policies which combine some of the singleton attributes. Secondly, in the schemes [10–12], data users are only allowed to search with a single keyword, which is not flexible enough. Although the scheme [7] supports recursive attribute set structure for more flexibility in specifying policies and the scheme [3] supports Boolean keyword search, none of the schemes support both of the two functionalities.

1.1. Our Contributions. In this paper, we present an attribute set-based Boolean keyword search (ASBBKS) scheme over encrypted PHR in cloud computing. In our scheme, each data user’s attributes are organized as recursive set structure, which enables more flexibility in user attribute organization and more efficiency in specifying policies than the existing ABKS schemes. Meanwhile, our scheme supports Boolean keyword search which is flexible in keyword expressivity. The ASBBKS model for encrypted PHR is shown in Figure 1. The main contributions of our paper are described in detail as follows:

- (i) In our scheme, each data user’s attributes can be organized as recursive set structure. That is, multiple values are assigned to an attribute in different sets. In the above example, the researcher’s attributes can be organized into a 2-depth recursive set structure as follows. For each role that a researcher has, a separate set of values {Department, Role} can be assigned.
 {Agency: College A, Role: Researcher,
 {Department: Medicine, Role: Director},
 {Department: Chemistry, Role: Professor}}.
- (ii) All authorized users can perform Boolean keyword search which is a more flexible search mechanism. For example, the PHR users can query the PHR which contains the keywords “(Treatment time: Jun 2021 \wedge Doctor name: Mike) \wedge (Drug name: Aspirin \vee Disease name: Neuralgia)” to the cloud server.
- (iii) We present the theoretical performance analysis of the ASBBKS scheme for computation and communication costs. In addition, the simulated results show that it is practical for the PHR systems.

1.2. Related Works. Searchable encryption (SE) was first proposed by Sont et al. [13] which enables users to implement keyword research over the encrypted data. There are

two categories for existing SE schemes: symmetric searchable encryption (SSE) [13–17] and public key encryption with keyword search (PEKS) [18–21]. SSE enables a user to encrypt his data and implement the keyword search by using his secret key. In a PEKS scheme, data owners can authorize search ability to each user by encrypting data with the user’s public key. Thus, each user can perform searches over encrypted data with his private key. However, the above two types of SE have a vulnerability that they do not support access control on data users. It means that each legitimate user can query all of the encrypted data, which may cause security issues.

To solve this problem, researchers have proposed SE schemes with access control such as attribute-based encryption with keyword search (ABKS) [22–26]. In such SE schemes, data owners can authorize search ability to data users whose attributes satisfy the access policy predefined by the data owner. There is a trusted authority in ABKS to generate public parameters. In addition, it also generates secret keys for data users. In [27], each user needs to submit his attribute structure and interested keywords to the trusted authority for generating trapdoors. Then, the data user can search over encrypted data with his trapdoors. However, it fails to preserve the privacy of interested keywords and they will be revealed to the trusted authority, which is not desired by the data user. Thus, Zheng et al. [28] proposed an improved SE scheme that makes use of an access tree to fulfill access control. In [28], each user sends an attribute set to the trusted authority for generating an attribute-related private key. The private key is used to generate trapdoors of interested keywords. Then, users can search over encrypted data with trapdoors without revealing interested keywords. However, in these schemes, user’s attributes are organized to a single set, which decreases the flexibility in specifying access policies.

In [29], the researchers proposed attribute set-based encryption (ASBE). In ASBE, the user’s attributes are organized in the form of recursive set structure which enables the data owners to encrypt data with more flexible access policies than ABE. Based on this technique, Xu et al. [7] proposed attribute set-based keyword search (ASBKS). Their scheme has better flexibility in user attribute organization and more efficiency in specifying policies. However, it only supports single keyword query, which limits the performance of searching.

To achieve multi-keyword search, Boolean keyword expression search was presented in [1]. In [1], keywords are divided into two parts: value and name, which are both organized in the form of an access tree structure. Inspired by this, we propose an ASBBKS scheme that can support Boolean keyword search, compounded attributes, and flexible search policies simultaneously.

1.3. Paper Organization. The rest of the paper is organized as follows. We introduce some preliminaries in Section 2. In Section 3, the concrete construction of the ASBBKS scheme is presented and the formal security proof of the ASBBKS scheme is provided. In Section 4, we evaluate the

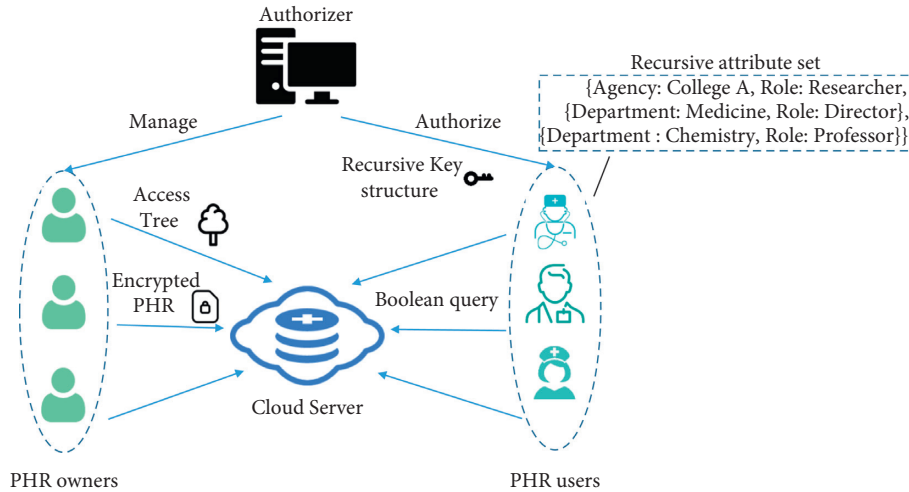


FIGURE 1: Our scheme model.

performance of our ASBBKS scheme and conduct simulation experiments to demonstrate its efficiency and practicality. Finally, we conclude the paper in Section 5.

2. Preliminaries

2.1. Bilinear Pairings. Let G_1 and G_2 represent the multiplicative cyclic group of order p , where p represents the prime number. Assume that g is the generator of group G_1 ; then, when $e: G_1 \times G_1 \rightarrow G_2$ satisfies the following conditions, we say it is a bilinear mapping [30]:

- (1) Bilinear: for any $g, h \in G_1$ and $a, b \in \mathbb{Z}_p$, $e(g^a, h^b) = e(g, h)^{ab}$.
- (2) Non-degenerate: $e(g, g) \neq 1$.

If there exists an efficient algorithm that can compute $e(g, h)$ for all $g, h \in G_1$, then G_1 is called a bilinear group.

2.2. Recursive Set Structure. We construct user's attributes as the recursive structure mentioned in [29]. In this structure, each element of a set can be an associated attribute or a set. They are organized like the tree structure with the notation of depth which limits this recursion. Here, we provide an example of this kind of set structure with depth 2 in Figure 2.

At the first layer, there are attribute elements and set elements. At the second layer, there are only attribute elements. For a set with depth 2, we can denote it as $\mathbb{A} = \{A_0, A_1, \dots, A_n\}$ where A_0 is the set of attributes at depth 1 and A_i is the i th set at depth 2 for $1 \leq i \leq n$.

2.3. Access Tree. In our scheme, the access tree structure is the same as that in [29], and Figure 3 shows an example of access tree \mathcal{T} . In \mathcal{T} , each inner node x represents a threshold gate which has a threshold value k_x . Assume that

the set of child nodes of x is represented by $\text{child}(x)$ and the number of child nodes is represented by n_x . In addition, each child node of x is labeled from left to right as 1 to n_x . Then, we have $1 \leq k_x \leq n_x$. When $k_x = 1$, the threshold gate transforms to a "OR" gate; when $k_x = n_x$, it transforms to an "AND" gate. In addition, each leaf node x of \mathcal{T} represents an attribute, denoted as $\text{att}(x)$. Furthermore, the parent of node x is presented as $\text{parent}(x)$, the label associated with x is presented as $\text{label}(x)$, the set of leaf nodes of \mathcal{T} is presented as $\text{lvs}(\mathcal{T})$, and the subtree of \mathcal{T} rooted at the node x is presented as \mathcal{T}_x .

For an attribute set structure $\mathbb{A} = \{A_0, A_1, \dots, A_n\}$ with depth 2 and an access tree \mathcal{T} , if at least one of the following conditions holds, we say that \mathbb{A} satisfies \mathcal{T} : (1) there is at least one subset of \mathbb{A} , in which the combination of its attributes satisfies \mathcal{T} ; (2) there are some translating nodes where the attributes from multiple sets in \mathbb{A} can be combined to satisfy \mathcal{T} . The translating node allows attributes from different sets to be combined to satisfy an access tree. Thus, by using some translating nodes, data users can combine attributes from multiple sets to satisfy an access tree. To make an explanation, we take the recursive set structure in Figure 2 and \mathcal{T} for example. In access tree \mathcal{T} , there are two inner nodes and one of them is a translating node. Their threshold gates are 1 and 2, respectively. For the attribute set structure \mathbb{A} in Figure 2, it can easily satisfy node v_1 . For node v_2 , subsets A_1 and A_2 can be combined together to satisfy it. However, if node v_2 is not a translating node, then there is no subset of \mathbb{A} that can satisfy node v_2 . Thus, using some translating nodes, data owners can selectively require users to combine attributes from either a single set or multiple sets to satisfy the access tree.

Secret share: for an access tree \mathcal{T} , each node x of \mathcal{T} with associated threshold k_x would equip with a polynomial q_x .

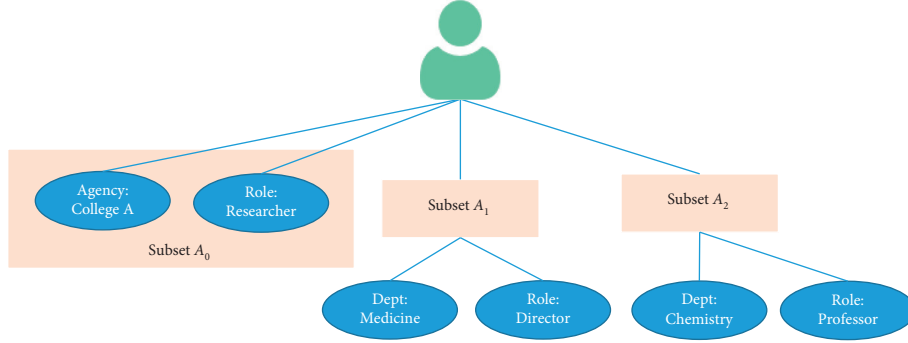


FIGURE 2: The recursive set structure.

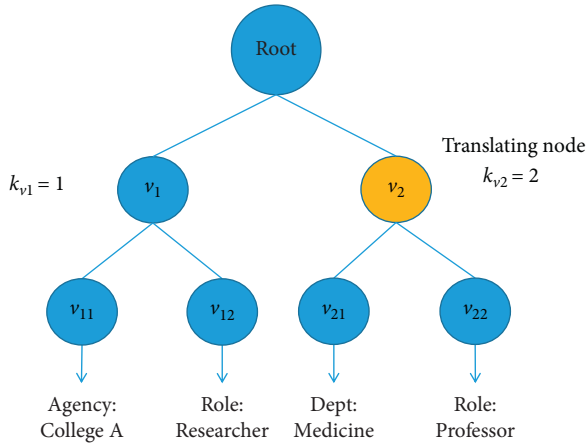


FIGURE 3: Access tree.

q_x is generated by running the secret share algorithm mentioned in [31]. This algorithm is used for distributing a secret s , and the process of this algorithm is as follows:

- (i) For the root node R , let $q_R(0) = s$ and then randomly select $k_R - 1$ coefficients for the polynomial q_R .
- (ii) For an inner node x , let $q_x(0) = q_{\text{parent}(x)}(\text{label}(x))$ and then randomly select $k_x - 1$ coefficients for the polynomial q_x .
- (iii) For a leaf node x , let the degree of q_x be $d_x = 0$ and $q_x(0) = q_{\text{parent}(x)}(\text{label}(x))$.

After executing the algorithm, the value $q_x(0)$ corresponding to each leaf node x becomes the secret share of s . We represent this algorithm as $\{q_x(0) \mid x \in \text{lvs}(\mathcal{T})\} \leftarrow \text{Share}(\mathcal{T}, s)$.

2.4. Boolean Query. In [3], a Boolean query requires that a server can find out all the encrypted data associated with an arbitrary Boolean expression among keywords. The expression can be denoted as $\text{Exp}(w_1, w_2, \dots, w_n)$. While calculating a Boolean expression, let $b_i = 1$ if the encrypted data contains w_i ; otherwise, set $b_i = 0$. Then, replace corresponding w_i in the expression with b_i . After that, if the result of the Boolean expression is “1,” encrypted data satisfy the condition of the Boolean query. In our scheme, keywords

are divided into two parts: name and value. They are organized as an access tree structure as shown in Figure 4, which is a more expressive searchable mechanism.

2.5. Definition of ASBBKS. In an ASBBKS scheme, there are several participants including multiple data owners and data users, a cloud server, and a trusted authority used for authorizing and managing data owners/users. An ASBBKS scheme consists of five algorithms: Setup, KeyGen, Encrypt, Trapdoor, and Test.

- (i) $\text{Setup}(k) \rightarrow \{\text{pk}, \text{mk}\}$: given the security parameter k , the Setup algorithm generates a public key pk and a master key mk .
- (ii) $\text{KeyGen}(\text{mk}, \mathbb{A}) \rightarrow \{\text{sk}\}$: the KeyGen algorithm uses the master key mk and the given attribute set structure \mathbb{A} to generate secret key sk corresponding to \mathbb{A} .
- (iii) $\text{Encrypt}(\text{pk}, W_V, \mathcal{T}) \rightarrow \{C\}$: the Encrypt algorithm inputs public key pk , a set of keyword values W_V , and an access tree \mathcal{T} and outputs the corresponding ciphertext C .
- (iv) $\text{Trapdoor}(\text{sk}, B_V) \rightarrow \{T\}$: the Trapdoor algorithm inputs secret key sk and a Boolean keyword value expression B_V , and outputs the corresponding trapdoor T .
- (v) $\text{Test}(C, T) \rightarrow \{0, 1\}$: the Test algorithm takes trapdoor T and ciphertext C as input. When the attribute set related to T matches with the access tree encrypted in C and a minimum subset W'_N of keyword names W_N encrypted in C satisfies the Boolean keyword expression B_V encrypted in T , it outputs 1. Otherwise, it outputs “ \perp ”.

In an ASBBKS scheme, the trusted authority firstly executes Setup to initialize the system parameters. Then, according to the user’s attribute set, it computes secret key for each user. Data owners should execute Encrypt on their data using preset access control policies to generate the corresponding ciphertexts and then upload them to the server. For data users, they can execute Trapdoor to generate trapdoors for searching over ciphertexts. In the end, the server executes Test to find all the data that are authorized to a user by the policies.

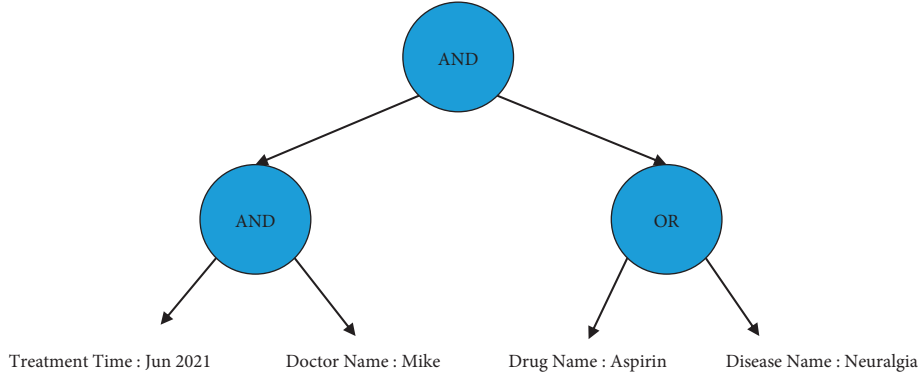


FIGURE 4: Boolean keyword expression.

2.6. *Security Model of ASBBKS.* We will give the security definition of ASBBKS. If there is no adversary who can win the following game with a non-negligible advantage in probabilistically polynomial time (PPT), then we can say that an ASBBKS scheme is secure against selectively chosen keyword attack (INDSCKA). In the following game, adversary \mathcal{A} interacts with the game challenger \mathcal{C} as follows.

- (i) **Setup:** in this stage, challenger \mathcal{C} first generates (pk, mk) through running $\text{Setup}(k)$. Then, pk will be sent to adversary \mathcal{A} .
- (ii) **Phase 1:** in this stage, adversary \mathcal{A} can make a polynomial number of queries as follows:
 - (1) **Secret key queries:** adversary \mathcal{A} adaptively queries secret keys for recursive attribute set structures to challenger \mathcal{C} .
 - (2) **Trapdoor queries:** adversary \mathcal{A} adaptively queries Boolean keyword value expressions B_V to challenger \mathcal{C} and gets the corresponding trapdoors.
- (iii) **Challenge:** after the above queries, adversary \mathcal{A} submits two distinct keyword value sets W_{V0} and W_{V1} to challenger \mathcal{C} . These sets cannot satisfy the Boolean keyword value expression B_V that is queried with the form of Trapdoor queries in Phase 1. Then, \mathcal{C} randomly selects a bit $\beta \in \{0, 1\}$, constructs the challenge ciphertext C^* of $W_{V\beta}$, and returns it to \mathcal{A} .
- (iv) **Phase 2:** adversary \mathcal{A} continues to make secret key queries and trapdoor queries similar to phase 1. It requires both W_{V0} and W_{V1} to not satisfy the Boolean keyword value expression B_V .
- (v) **Guess:** adversary \mathcal{A} has to guess and output a bit $\beta' \in \{0, 1\}$. It wins the game if $\beta' = \beta$.

3. Our Construction of ASBBKS

In this section, we will provide a concrete construction of our scheme and the formal security proof. In the following construction and security proof, we assume the depth of the user's attribute set structure to be 2.

3.1. The Concrete Construction

3.1.1. *Setup* $(\kappa) \rightarrow \{pk, mk\}$. Let $e: G \times G \rightarrow G_T$ be a bilinear pairing in which G and G_T represent two cyclic groups of prime order p . Assume that g is the generator of G and $H_0: \{0, 1\}^* \rightarrow Z_p^*$ and $H_1: \{0, 1\}^* \rightarrow G$ are two collision-resistant hash functions. Next, it randomly selects $\beta_1, \beta_2, \alpha \in Z_p^*$ and calculates $h_1 = g^{\beta_1}$, $h_2 = g^{\beta_2}$, $h_3 = g^\alpha$. The master key and the public key (pk, mk) are set to be

$$\begin{aligned} pk &= \langle G, G_T, p, g, e, h_1, h_2, h_3, H_0(\cdot), H_1(\cdot) \rangle, \\ mk &= \langle \beta_1, \beta_2, \alpha \rangle. \end{aligned} \quad (1)$$

3.1.2. *KeyGen* $(\mathbb{A}, mk) \rightarrow \{sk\}$. The input recursive attribute set structure is parsed as $\mathbb{A} = \{A_0, A_1, \dots, A_n\}$. Assume that each A_i has m_i attribute members; then, there is $A_i = \{a_{i,1}, \dots, a_{i,m_i}\}$ for $0 \leq i \leq n$. Firstly, it randomly selects a value $r \in Z_p^*$ for attribute set structure \mathbb{A} . For each subset A_i , it randomly selects n values $\{r_i \in Z_p^* \mid 1 \leq i \leq n\}$ and sets $r_0 = r$. Furthermore, it randomly selects a set of values $\{r_{i,j} \in Z_p^* \mid 0 \leq i \leq n, 1 \leq j \leq m_i\}$ for each attribute. Finally, it calculates $B = g^{(\alpha-r)/\beta_1}$, $B_{i,j} = g^{r_i} H_1(a_{i,j})^{r_{i,j}}$ and $B'_{i,j} = g^{r_{i,j}}$ for $0 \leq i \leq n, 1 \leq j \leq m_i$ and $D_i = g^{(r+r_i)/\beta_2}$ for $1 \leq i \leq n$. The secret key for \mathbb{A} is set to be

$$\begin{aligned} sk &= \langle \mathbb{A}, B, \{(B_{i,j}, B'_{i,j}) \mid 0 \leq i \leq n, 1 \leq j \leq m_i\}, \\ &\quad \{D_i \mid 1 \leq i \leq n\} \rangle. \end{aligned} \quad (2)$$

At the translating node, each element D_i in a secret key supporting r_i of set A_i at depth 2 translates to r of set A_0 at depth 1. Elements D_i and $D_{i'}$ can be combined as $D_i/D_{i'}$ to translate $r_{i'}$ to r_i at the translating nodes.

3.1.3. *Encrypt* $(pk, W_V, \mathcal{T}) \rightarrow \{C\}$. $W_V = (w_{\rho(1)}, w_{\rho(2)}, \dots, w_{\rho(m)})$ is a set of keyword values and $W_N = (\rho(1), \rho(2), \dots, \rho(m))$ is the set of the keyword names. This algorithm randomly selects $m+1$ values $s_0, s_1, \dots, s_m \in Z_p^*$ and computes $C_1 = h_1^{s_0}$, $C_2 = h_2^{s_0}$, $C_{i,1} = g^{s_i}$, $C_{i,2} = h_3^{(s_0+s_i)} \cdot h_2^{H_0(w_{\rho(i)})s_i}$ for

$1 \leq i \leq m$. After that, it computes secret shares of s_0 through implementing $\{q_v(0) | v \in \text{lvs}(\mathcal{T})\} \leftarrow \text{Share}(\mathcal{T}, s_0)$. Further, for each $v \in \text{lvs}(\mathcal{T})$, it computes $C_v = g^{q_v(0)}$ and $C'_v = H_1(\text{att}(v))^{q_v(0)}$. We assume that the set of translating nodes in \mathcal{T} is represented as $\text{trans}(\mathcal{T})$. Then, it calculates $\tilde{C}_x = h_2^{q_x(0)}$ for each $x \in \text{trans}(\mathcal{T})$. Finally, the ciphertext is set to be

$$C = \langle \mathcal{T}, W_N, C_1, C_2, \{(C_{i,1}, C_{i,2}) | 1 \leq i \leq m\}, \{(C_v, C'_v) | v \in \text{lvs}(\mathcal{T})\}, \{\tilde{C}_x | x \in \text{trans}(\mathcal{T})\} \rangle. \quad (3)$$

In user keys, elements (\tilde{C}_x) 's and (D_i) 's support translation between sets at a translating node x . We will describe it later in the Test algorithm.

3.1.4. Trapdoor $(sk, \mathcal{B}_V) \rightarrow \{T\}$. \mathcal{B}_V represents the Boolean keyword value expression which is an access tree. \mathcal{B}_N represents the Boolean keyword name expression which is an access tree with the same structure as \mathcal{B}_V . Taking sk and \mathcal{B}_V as inputs, it randomly selects a value $t \in Z_p^*$ and calculates the secret shares of t by implementing $\{q_v(0) | v \in \text{lvs}(\mathcal{B}_N)\} \leftarrow \text{Share}(\mathcal{B}_N, t)$. Then, this algorithm calculates $T_{\rho(v),1} = (h_3 h_2^{(H_0 \bar{w}_{\rho(v)})})^{q_v(0)}$ and $T_{\rho(v),2} = g^{\hat{q}_v(0)}$. Parsing sk as $\langle \mathbb{A}, B, \{(B_{i,j}, B'_{i,j}) | 0 \leq i \leq n, 1 \leq j \leq m_i\}, \{D_i | 1 \leq i \leq n\} \rangle$, it further computes $\bar{B} = B^t$, $\bar{B}_{i,j} = B_{i,j}^t$, $\bar{B}'_{i,j} = B'_{i,j}^t$ for $0 \leq i \leq n, 1 \leq j \leq m_i$ and $\bar{D}_i = D_i^t$ for $1 \leq i \leq n$. Finally, the trapdoor for \mathcal{B}_V is

$$T = \langle \mathbb{A}, \mathcal{B}_N, \{(T_{\rho(v),1}, T_{\rho(v),2}) | v \in \text{lvs}(\mathcal{B}_N)\}, \bar{B}, \{(\bar{B}_{i,j}, \bar{B}'_{i,j}) | 0 \leq i \leq n, 1 \leq j \leq m_i\}, \{\bar{D}_i | 1 \leq i \leq n\} \rangle. \quad (4)$$

3.1.5. Test $(C, T) \rightarrow \{0, 1\}$. Cloud server takes input ciphertext $C = \langle \mathcal{T}, W_N, C_1, C_2, \{(C_{i,1}, C_{i,2}) | 1 \leq i \leq m\}, \{(C_v, C'_v) | v \in \mathbb{V}\}, \{\tilde{C}_x | x \in \text{trans}(\mathcal{T})\} \rangle$ and the trapdoor $T = \langle \mathbb{A}, \mathcal{B}_N, \{(T_{\rho(v),1}, T_{\rho(v),2}) | v \in \text{lvs}(\mathcal{B}_N)\}, \bar{B}, \{(\bar{B}_{i,j}, \bar{B}'_{i,j}) | 0 \leq i \leq n, 1 \leq j \leq m_i\}, \{\bar{D}_i | 1 \leq i \leq n\} \rangle$. The complete Test algorithm consists of the following 3 steps.

Step 1. In this step, for the given access tree \mathcal{T} and key structure \mathbb{A} , it returns a set S_τ . The elements of this set are some labels for each node τ within \mathcal{T} . Each label t in S_τ represents a set A_t and each set satisfies the subtree \mathcal{T}_τ . There is $\mathcal{T}_R = \mathcal{T}$ for the root node R and the related set is S_R . If \mathbb{A} does not satisfy \mathcal{T} , the return of this algorithm is "0." Otherwise, for node τ , it picks one label from the set S_τ , denoted as i . Then, it executes a recursive function $\text{DecryptNode}(C, T, \tau, i)$, which will return F_τ as the result. According to the type of node τ , the function $\text{DecryptNode}(C, T, \tau, i)$ will be computed in two different ways.

- (1) When τ is a leaf node: if $\text{att}(\tau) \notin A_i$, it returns "1". Otherwise, we have

$$\begin{aligned} \text{DecryptNode}(C, T, \tau, i) &= \frac{e(C_\tau, \bar{B}_{i,j})}{e(C'_\tau, \bar{B}'_{i,j})} \\ &= e(g, g)^{t \cdot r_i \cdot q_\tau(0)}. \end{aligned} \quad (5)$$

- (2) When τ is not a leaf node:

- (i) Firstly, it calculates a set E_τ , which is composed of k_τ child nodes of τ . In E_τ , each node z must satisfy one of the following two cases: (1) label $i \in S_z$ and (2) z is a translating node and there exists a label i' that satisfies $i' \in S_z$ and $i' \neq i$. If such a set does not exist, it returns "1".
- (ii) Execute $\text{DecryptNode}(C, T, z, i)$ for each node $z \in E_\tau$ which satisfies label $i \in S_z$ and then return F_z as the result.
- (iii) Execute $\text{DecryptNode}(C, T, z, i')$ for each translating node $z \in E_\tau$ which satisfies $i' \in S_z$ and $i' \neq i$ and return F'_z as the result. If $i = 0$, elements $\bar{D}_{i'}$ and \tilde{C}_z can be used to translate F'_z to F_z .

$$\begin{aligned} F_z &= \frac{e(\bar{D}_{i'}, \tilde{C}_z)}{F'_z} \\ &= \frac{e\left(g^{t(r+r_i)}/\beta_2, g^{\beta_2 \cdot q_z(0)}\right)}{e(g, g)^{t \cdot r_i' \cdot q_z(0)}} \\ &= e(g, g)^{t \cdot r \cdot q_z(0)}. \end{aligned} \quad (6)$$

If $i \neq 0$, elements \bar{D}_i and $\bar{D}_{i'}$ together with \tilde{C}_z can be used to translate F'_z to F_z .

$$\begin{aligned} F_z &= e\left(\frac{\bar{D}_i}{\bar{D}_{i'}}, \tilde{C}_z\right) \cdot F'_z \\ &= e\left(g^{\frac{t(r_i - r_i')}{\beta_2}}, g^{\beta_2 \cdot q_z(0)}\right) e(g, g)^{t \cdot r_i' \cdot q_z(0)} \\ &= e(g, g)^{t \cdot r_i \cdot q_z(0)}. \end{aligned} \quad (7)$$

- (iv) After computing F_z for each node z in E_τ , it computes F_τ as follows:

$$F_\tau = \prod_{z \in E_\tau} F_z^{\Delta_{i,U_z}(0)} = \begin{cases} e(g, g)^{t \cdot r \cdot q_\tau(0)}, & i = 0 \\ e(g, g)^{t \cdot r_i \cdot q_\tau(0)}, & i \neq 0 \end{cases}, \quad (8)$$

where $\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} (x - j) / (i - j)$, $v = \text{label}(z)$, and $U_z = \{\text{label}(z) : z \in E_\tau\}$.

After the above steps, it further computes the function $\text{DecryptNode}(C, T, R, i)$ for root node R and returns F_R as the result as follows.

$$F_R = \begin{cases} e(g, g)^{t \cdot r \cdot s_0}, & i = 0, \\ e(g, g)^{t \cdot r_i \cdot s_0}, & i \neq 0, \end{cases} \quad (9)$$

In the end, it computes a value F . When $i = 0$, it assumes $F = F_R$. When $i \neq 0$, it computes F as follows:

$$F = \frac{e(\overline{D}_i, C_3)}{F_R} = \frac{e(g, g)^{t \cdot (r+r_i) \cdot s_0}}{e(g, g)^{t \cdot r_i \cdot s_0}} = e(g, g)^{t r s_0}. \quad (10)$$

Step 2. Firstly, it computes a value L with the following formula:

$$L = e(\overline{B}, C_1) = e(g^{t(\alpha-r)/\beta_1}, g^{\beta_1 s_0}) = e(g, g)^{t(\alpha-r)s_0}. \quad (11)$$

Then, it computes $F \cdot L$ with the following formula:

$$F \cdot L = e(g, g)^{t r s_0} e(g, g)^{t(\alpha-r)s_0} = e(g, g)^{t \alpha s_0}. \quad (12)$$

Step 3. It selects a minimum subset W'_N from the set of keyword names W_N , which satisfies the Boolean keyword name expression \mathcal{B}_N . If W'_N does not exist, it returns "0." If node \hat{x} is a leaf node of access tree \mathcal{B}_N and is related to the search token T , then the keyword name associated with this node is denoted by $\rho(\hat{x})$. Further, for each keyword name $\rho(\hat{x}) \in W'_N$, it computes

$$E_{\hat{x}} = \frac{e(C_{\rho(\hat{x}),2}, T_{\rho(\hat{x}),2})}{e(T_{\rho(\hat{x}),1}, C_{\rho(\hat{x}),1})} = \frac{e\left(h_3^{s_0+s_{\rho(\hat{x})}} h_2^{H_0(w_{\rho(\hat{x})}^{s_{\rho(\hat{x})}})}, g^{\hat{q}_x^{(0)}}\right)}{e\left(\left(h_3 h_2^{H_0(\overline{w}_{\rho(\hat{x})})}\right)^{\hat{q}_x^{(0)}}, g^{s_{\rho(\hat{x})}}\right)}. \quad (13)$$

If $w_{\rho(\hat{x})} = \overline{w}_{\rho(\hat{x})}$, then there is $E_{\hat{x}} = e(g, g)^{\alpha s_0 \hat{q}_x^{(0)}}$.

When node \hat{x} is not a leaf node, for all child nodes \hat{z} of \hat{x} , assume that $S_{\hat{x}}$ represents an arbitrary set with size $k_{\hat{x}}$ consisting of children nodes \hat{z} and $E_{\hat{z}} \neq \perp$. If $S_{\hat{x}}$ does not exist, then $E_{\hat{z}} \neq \perp$; otherwise, it utilizes the polynomial interpolation to calculate $E_{\hat{x}}$ to get

$$E_{\hat{x}} = e(g, g)^{\alpha s_0 \hat{q}_x^{(0)}}. \quad (14)$$

At last, for the root node of \mathcal{B}_N , it computes $E_{\hat{R}}$ and checks whether the following equation holds:

$$E_{\hat{R}} = F \cdot L. \quad (15)$$

If the equation above does not hold, then the algorithm would keep finding another subset of keyword names from the set of keyword names W_N which satisfies \mathcal{B}_N and repeat the checking as above. If there exists no such keyword name subset such that the above equation holds, it returns "1".

We depict the whole processing steps of our scheme in Figure 5.

Remark 1. For the user's attribute set structure with depth d , for each level i , a value β_i needs to be selected. β_i supports the translations between sets at level i or between a set at level i and its outer set at level $i - 1$. Translations across multiple levels will use corresponding translating values and different β s.

3.2. Security Proof. In this section, we will provide the formal proof of Theorem 1 to prove that our scheme is secure.

Theorem 1. *The above scheme is selectively secure against chosen keyword attack in the generic bilinear group model.*

Proof. In our security model, the adversary \mathcal{A} assumes to distinguish $g^{\alpha(s_0+s_i)} g^{\beta_2 H_0(w_{\rho(i)}^0)^{s_i}}$ from g^θ and $g^{\alpha(s_0+s_i)} g^{\beta_2 H_0(w_{\rho(i)}^1)^{s_i}}$ from g^θ , where θ is randomly selected from Z_p^* , W_0 and W_1 are two different keyword sets, $w_{\rho(i)}^0 \in W_0$, and $w_{\rho(i)}^1 \in W_1$. Since \mathcal{A} has the same probability to distinguish both of them, it is easy to distinguish $g^{\beta_2 H_0(w_{\rho(i)}^0)^{s_i}}$ from g^θ . That is, this game can be transformed to \mathcal{A} that has the advantage $\epsilon/2$ to distinguish $g^{\alpha(s_0+s_i)}$ from g^θ .

The challenger \mathcal{C} can be constructed in this game as follows.

- (i) **Setup:** the challenger \mathcal{C} selects parameters $\alpha, \beta_1, \beta_2 \in Z_p^*$ and sends public parameter $\text{pk} = \langle G, G_T, p, g, e, g^{\beta_1}, g^{\beta_2}, g^\alpha, H_0 \rangle$ to \mathcal{A} . After that, \mathcal{A} selects a challenge access tree \mathcal{T}^* and returns it to \mathcal{C} .
- (ii) **Hash₁-Queries:** challenger \mathcal{C} would maintain an H -list, which is empty at first. Given an input attribute $a_{i,j}$, if $a_{i,j}$ has not been queried, it randomly selects $t_{i,j} \in Z_p^*$ and then returns $g^{t_{i,j}}$ to \mathcal{A} and the tuple $(a_{i,j}, t_{i,j})$ will be added to H -list. For those attributes $a_{i,j}$ that have been queried, \mathcal{C} will directly return $g^{t_{i,j}}$ to \mathcal{A} . \mathcal{A} can query this random oracle for polynomially many times. \square

Phase 1. Adversary \mathcal{A} can make the secret key and trapdoor queries for polynomially many times.

- (i) **Secret key queries:** firstly, for an attribute set structure $\mathbb{A} = \{A_0, A_1, \dots, A_n\}$ where $A_i = \{a_{i,1}, \dots, a_{i,m_i}\}$ for $0 \leq i \leq n$, \mathcal{C} will randomly select a value $r \in Z_p^*$ for \mathbb{A} and compute $B = g^{(\alpha-r)/\beta_1}$. For each subset A_i , it will randomly select $r_i \in Z_p^*$ and a value $r_{i,j} \in Z_p^*$ for $0 \leq i \leq n, 1 \leq j \leq m_i$. Next, use these parameters to compute corresponding $B_{i,j} = g^{r_i} g^{t_{i,j} r_{i,j}}, B_{i,j}' = g^{r_{i,j}}$, and $D_i = g^{(r+r_i)/\beta_2}$. Finally, \mathcal{C} constructs the secret key $\text{sk} = \langle \mathbb{A}, B, \{(B_{i,j}, B_{i,j}') \mid 0 \leq i \leq n, 1 \leq j \leq m_i\}, \{D_i \mid 1 \leq i \leq n\} \rangle$ with the above parameters and returns it to \mathcal{A} .
- (ii) **Trapdoor queries:** if \mathcal{A} has queried secret key sk , then \mathcal{C} directly runs the Trapdoor algorithm to get

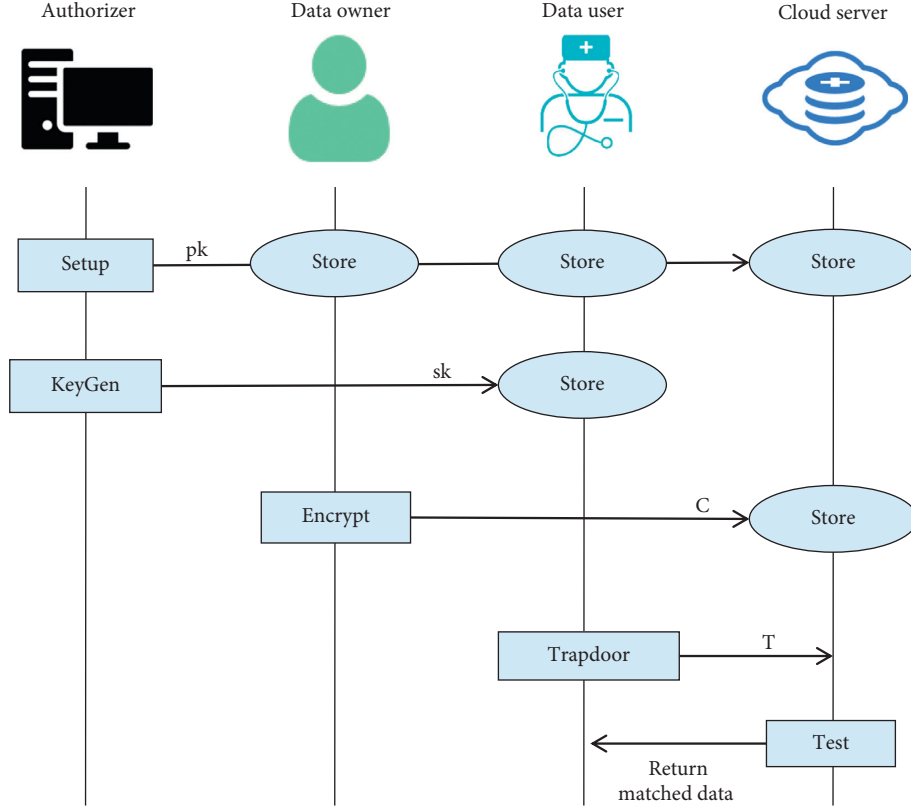


FIGURE 5: Workflow of our scheme.

the trapdoor T . Challenger \mathcal{C} randomly selects $t \in Z_p^*$ and computes the secret shares of t by running $\{\hat{q}_v(0) \mid v \in \text{lvs}(\mathcal{B}_N)\} \leftarrow \text{Share}(\mathcal{B}_N, t)$ and computes corresponding $T_{\rho(v),1} = (g^\alpha g^{\beta_2 H_0} (\bar{w}_{\rho(v)})) \hat{q}_v(0)$, $T_{\rho(v),2} = \hat{g}^{\hat{q}_v(0)}$, $\bar{B} = B^t$, $\bar{B}_{i,j} = B_{i,j}^t$, $\bar{B}'_{i,j} = B'_{i,j}^t$, and $\bar{D}_i = D_i^t$. Then the trapdoor for Boolean keyword value expression \mathcal{B}_V is $T = \langle \mathbb{A}, \mathcal{B}_N, \{(T_{\rho(v),1}, T_{\rho(v),2}) \mid v \in \text{lvs}(\mathcal{B}_N)\}, \bar{B}, \{\bar{B}_{i,j}, \bar{B}'_{i,j} \mid 0 \leq i \leq n, 1 \leq j \leq m_i\}, \{\bar{D}_i \mid 1 \leq i \leq n\} \rangle$, and the challenger \mathcal{C} returns it to \mathcal{A} .

Challenge: \mathcal{A} submits two distinct keyword value sets $W_{V0} = (w_{\rho(1)}^0, w_{\rho(2)}^0, \dots, w_{\rho(m)}^0)$ and $W_{V1} = (w_{\rho(1)}^1, w_{\rho(2)}^1, \dots, w_{\rho(m)}^1)$ to \mathcal{C} , and these sets are of equal size. Then, let $W_N = (\rho(1), \rho(2), \dots, \rho(m))$ represent the set of keyword name. For these two sets, they cannot satisfy the Boolean keyword value expression B_V . Then, \mathcal{C} randomly selects $s_0, s_i \in Z_p^*$, for $1 \leq i \leq m$ and computes secret shares of s_0 by executing $\text{Share}(\mathcal{T}^*, s_0) \rightarrow \{q_v(0) \mid v \in \text{lvs}(\mathcal{T}^*)\}$.

Then, \mathcal{C} randomly selects a bit $\beta \in \{0, 1\}$ and constructs the challenge ciphertext $C^* = \langle \mathcal{T}^*, W_N, C_1, C_2, \{(C_{i,1}, C_{i,2}) \mid 1 \leq i \leq m\}, \{(C_v, C'_v) \mid v \in \text{lvs}(\mathcal{T})\}, \{\tilde{C}_x \mid x \in \text{trans}(\mathcal{T}^*)\} \rangle$, and then \mathcal{C} returns it to \mathcal{A} .

- (i) If $\beta = 0$, it randomly selects $\theta \in Z_p^*$ and outputs $C_1 = g^{\beta_1 s_0}, C_2 = g^{\beta_2 s_0}$, for $1 \leq i \leq m$, $C_{i,1} = g^{s_i}, C_{i,2} = g^\theta$, $\{(C_v = g^{q_v(0)}, C'_v = g^{t_{i,j} q_v(0)} \mid v \in \text{lvs}(\mathcal{T}^*), \text{att}(v) = a_{i,j})\}$, $\{\tilde{C}_x = g^{\beta_2 q_x(0)} \mid x \in \text{trans}(\mathcal{T}^*)\}$
- (ii) If $\beta = 1$, it outputs $C_1 = g^{\beta_1 s_0}, C_2 = g^{\beta_2 s_0}$, for $1 \leq i \leq m$, $C_{i,1} = g^{s_i}, C_{i,2} = g^{\alpha(s_0 + s_i)}$, $\{(C_v = g^{q_v(0)}, C'_v = g^{t_{i,j} q_v(0)} \mid v \in \text{lvs}(\mathcal{T}^*), \text{att}(v) = a_{i,j})\}$, $\{\tilde{C}_x = g^{\beta_2 q_x(0)} \mid x \in \text{trans}(\mathcal{T}^*)\}$.

Phase 2. After receiving the challenge ciphertext C^* , \mathcal{A} can perform secret key queries and trapdoor queries mentioned in phase 1. It requires both W_{V0} and W_{V1} to not satisfy the Boolean keyword value expression B_V .

Guess: \mathcal{A} outputs its guess, which requires \mathcal{A} to distinguish $g^{(s_0 + s_i)}$ from g^θ to win this game.

Analysis: if \mathcal{A} can construct $e(g^\eta, g^{\alpha(s_0 + s_i)})$ for some g^η , \mathcal{A} has the ability to distinguish $g^{\alpha(s_0 + s_i)}$ from a random element g^θ . Thus, we continue to analyze that adversary \mathcal{A} constructs $e(g, g)^{\eta \alpha(s_1 + s_2)}$ for some g^η with negligible advantage. That means \mathcal{A} cannot win our game with a non-negligible advantage.

From the above phases, it can be easily found that s_i and α have appeared, so \mathcal{A} only needs to construct αs_0 . With

terms $g^{\beta_1 s_0}$ and $g^{(\alpha-r)/\beta_1}$, \mathcal{A} can construct $e(g, g)^{\alpha s_0} e(g, g)^{-r s_0}$ and then \mathcal{A} needs to conceal $e(g, g)^{r s_0}$. With terms $g^{\beta_2 s_0}$ and $g^{(r+r_i)/\beta_2}$, \mathcal{A} can construct $e(g, g)^{(\alpha s_0 + r_i s_0)}$ and \mathcal{A} needs to conceal $e(g, g)^{r_i s_0}$. Then, $g^{r_i s_0}$ should be constructed which will use terms $g^{r_i + a_{ij} r_i}$, $g^{r_{ij}}$, $g^{q_v(0)}$, and $g^{a_{i,j} q_v(0)}$ because $q_v(0)$ is the secret share of s_0 . However, $g^{r_i s_0}$ cannot be constructed since there are not enough attribute values $g^{q_v(0)}$ for \mathcal{A} to satisfy the access tree \mathcal{T} .

4. Performance Analysis

In the existing ABKS schemes, no scheme can support both Boolean keyword search and recursive set structure. At first, we compare the existing schemes [1, 7, 28] in Table 1 with our scheme in the aspect of their functionalities. We assume four parts to compare, search method, access structure, attribute set structure, and translating nodes. From Table 1, we can conclude that our scheme is the first one that supports compound attributes, flexible access policies' specifying, and Boolean keyword search simultaneously.

4.1. Theoretical Analysis. In our scheme, three operations are the most time consuming which are, respectively, the bilinear pairing, the modular exponentiation, and the hash function H_1 . Since H_1 can be precomputed, we just consider the former two operations in the following analysis.

In theoretical analysis, we focus on the computation complexity and storage overhead of each step. Firstly, we assume a user's 2-level recursive attribute set to be $\mathbb{A} = \{A_0, A_1, \dots, A_n\}$ where $A_i = \{a_{i,1}, a_{i,2}, \dots, a_{i,m_i}\}$. Let $M = m_0 + m_1 + \dots + m_n$. Then, let the leaf nodes of an access tree \mathcal{T} be $|\text{lvs}(\mathcal{T})|$ and translating nodes be $|\text{trans}(\mathcal{T})|$. For the keyword set used in Encrypt, we denote the number of keywords in this set to be N and the leaf nodes of a Boolean keyword value expression B_V to be $|\text{lvs}(B_V)|$. We organize the theoretical computation complexity and storage overhead of the existing schemes [1, 7, 28] and our scheme in Table 2.

KeyGen. For the KeyGen algorithm in our scheme, its computation complexity is $2M + 2n + 1$ exponentiations in G . The storage overhead is $2M + n + 1$ group elements in G . In [28], the KeyGen algorithm takes $2M + 1$ exponentiations in G . The storage overhead is $2M$ group elements in G . In [7], the computation complexity of its KeyGen algorithm is $1 + 2n + 2M$ exponentiations in G . The storage overhead is $2M + n + 1$ group elements in G . In [1], the KeyGen algorithm takes $2 + 2n$ exponentiations in G . The storage takes $2M + 1$ group elements in G .

Encrypt. For the Encrypt algorithm in our scheme, its computation complexity is $|\text{trans}(\mathcal{T})| + 2|\text{lvs}(\mathcal{T})| + 3N + 2$ exponentiations in G . The storage overhead is $|\text{trans}(\mathcal{T})| + 2|\text{lvs}(\mathcal{T})| + 2N + 2$ group elements in G . In [28], the Encrypt algorithm takes $2|\text{lvs}(\mathcal{T})| + 4$ exponentiations in G . The storage overhead is $2|\text{lvs}(\mathcal{T})| + 3$ group elements in G .

TABLE 1: Comparison of functionalities between SE schemes.

	Boolean keyword search	Access structure	Compound attributes	Translating nodes
[1]	✓	LSSS	×	×
[28]	×	Access tree	×	×
[7]	×	Access tree	✓	✓
Our ASBBKS	✓	Access tree	✓	✓

In [28], the computation complexity of its Encrypt algorithm is $|\text{trans}(\mathcal{T})| + 2|\text{lvs}(\mathcal{T})| + 5$ exponentiations in G . The storage overhead is $|\text{trans}(\mathcal{T})| + 2|\text{lvs}(\mathcal{T})| + 4$ group elements in G . In [28], the Encrypt algorithm takes $1 + 2|\text{lvs}(\mathcal{T})| + 3N$ exponentiations in G . The storage takes $1 + 2|\text{lvs}(\mathcal{T})| + 2N$ group elements in G .

Trapdoor. For the Trapdoor algorithm in our scheme, its computation complexity is $2M + n + 3|\text{lvs}(B_N)| + 1$ exponentiations in G . The storage overhead is $2M + n + 1 + 2|\text{lvs}(B_N)|$ group elements in G . In [1], the Trapdoor algorithm takes $2M + 4$ exponentiations in G . The storage overhead is $2M + 3$ group elements in G . In [28], the computation complexity of its Trapdoor algorithm is $4 + n + 2M$ exponentiations in G . The storage overhead is $3 + n + 2M$ group elements in G . In [7], the Trapdoor algorithm takes $1 + 2M + 3|\text{lvs}(B_N)|$ exponentiations in G . The storage takes $1 + 2M + 2|\text{lvs}(B_N)|$ group elements in G .

Test. For the Test algorithm in our scheme, we first assume \mathbb{A} has l leaf nodes satisfying \mathcal{T} and k_i translating nodes on the path from i th leaf node used to the root node where $1 \leq i \leq l$, and let $k = k_1 + k_2 + \dots + k_l$. In addition, we denote the number of all used nodes in \mathcal{T} as t , and the computation complexity is $2l + k + 1 + 2|\text{lvs}(B_N)|$ pairings and $t - 1 + |\text{lvs}(B_N)|$ exponentiations in G_T . The storage overhead depends on the number of the matched ciphertexts, which we denote as NC. In [28], the Test algorithm takes $2l + 3$ pairings and $t - 1$ exponentiations in G_T and the storage overhead we assume is NC. In [7] the computation complexity of its Test algorithm is $3 + k + 2l$ pairings and $t - 1$ exponentiations in G_T , and the storage overhead we assume is NC. In [1], the Search algorithm takes $1 + 2l + 2|\text{lvs}(B_N)|$ pairings and $|\text{lvs}(\mathcal{T})| + |\text{lvs}(B_N)|$ exponentiations in G_T and the storage overhead we assume is NC.

Remark 2. From the analysis, the difference in computation cost between our scheme and [28] depends linearly on the number of keywords, the number of inner sets in \mathbb{A} , and the number of translating nodes in \mathcal{T} . When there is only one set in \mathbb{A} , one keyword in the Encrypt algorithm and Trapdoor algorithm, and no translating node in access tree \mathcal{T} , the computation cost of our scheme is the same as that of the scheme in [28]. The difference in computation cost between our scheme and [7] depends

TABLE 2: Theoretical efficiency analysis and comparison between schemes.

	[31]	[28]	[11]	Our scheme
KeyGen	$(2M + 1)E$	$(1 + 2n + 2M)E$	$(2 + 2n)E$	$(1 + 2n + 2M)E$
Encrypt	$(2 vs T + 4)E$	$(trans(T) + 2 vs T + 5)E$	$(1 + 2 vs T + 3N)E$	$(trans(T) + 2 vs T + 3N + 2)E$
Trapdoor	$(2M + 4)E$	$(4 + n + 2M)E$	$(1 + 2M + 3 vs(B_N))E$	$(1 + 3 vs(B_N) + n + 2M)E$
Test	$(2l + 3)e + (t - 1)E_T$	$(2l + k + 3)e + (t - 1)E_T$	$(1 + 2l + 2 vs(B_N))e + (vs T + vs(B_N))E_T$	$(2l + k + 1 + 2 vs(B_N))e + (t - 1 + vs(B_N))E_T$
KeyGen	$(2M)G$	$(1 + n + 2M)G$	$(1 + 2M)G$	$(1 + n + 2M)G$
Encrypt	$(2 vs T + 3)G$	$(2 vs T + trans(T) + 4)G$	$(1 + 2 vs T + 2N)G$	$(2 vs T + trans(T) + 2N + 2)G$
Trapdoor	$(2M + 3)G$	$(3 + n + 2M)G$	$(1 + 2M + 2 vs(B_N))G$	$(1 + n + 2M + 2 vs(B_N))G$
Test	NC	NC	NC	NC

e : evaluation of a bilinear pairing; E : evaluation of a modular exponentiation in G ; E_T : evaluation of a modular exponentiation in G_T ; M : number of attributes in \mathbb{A} ; n : number of sets in \mathbb{A} ; l : number of attributes used in \mathbb{A} required to satisfy T ; t : number of all used nodes in T ; $|vs(T)|$: number of leaf nodes used in T ; $|vs(B_N)|$: number of keywords used in B_N ; $|trans(T)|$: number of translating nodes used in T ; k : number of all translating nodes on the path from each leaf node used to the root.

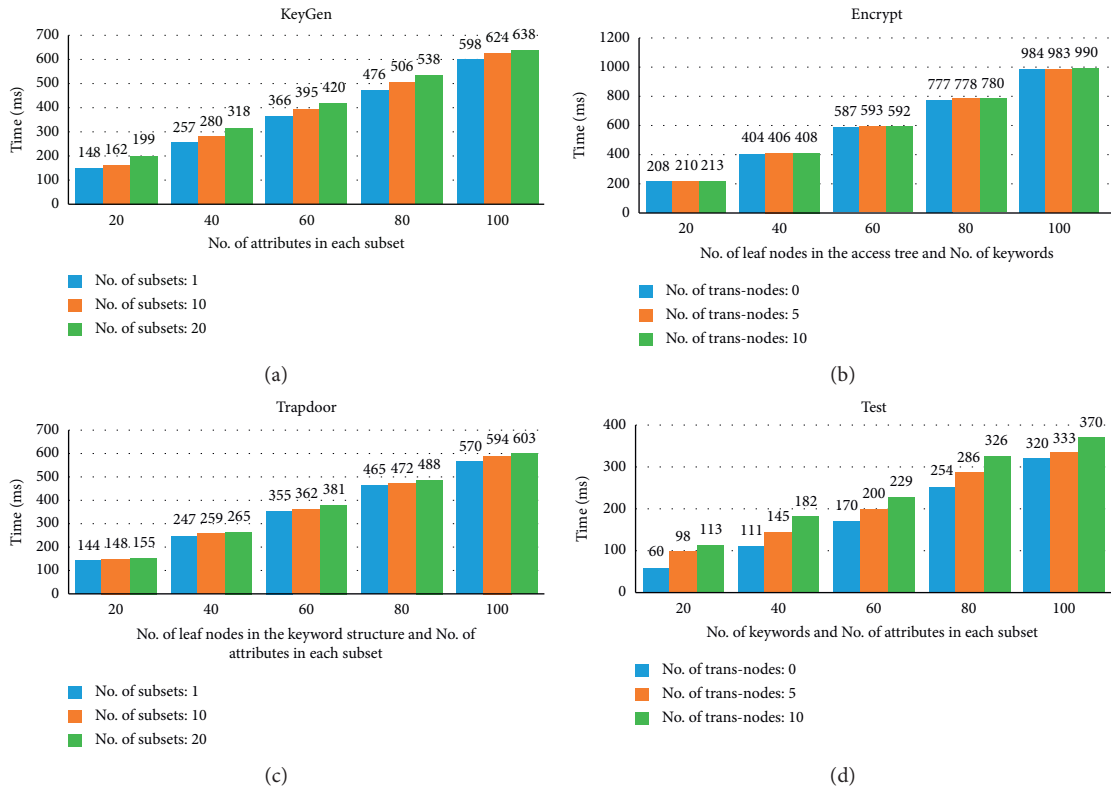


FIGURE 6: Operation time of the algorithms in ASBBKS. (a) KeyGen. (b) Encrypt. (c) Trapdoor. (d) Test.

linearly on the number of keywords. When there is only one keyword in the Encrypt algorithm and Trapdoor algorithm, the computation cost of our scheme is the same as that of the scheme in [7]. The difference in computation cost between our scheme and [1] depends linearly on the number of inner sets in \mathbb{A} and the number of translating nodes in \mathcal{T} . When there is one set in \mathbb{A} and no translating node in access tree \mathcal{T} , the computation cost of our scheme is the same as that of the scheme in [1]. In summary, our ASBBKS scheme supports compound attributes and Boolean queries by adding some computation operations, which is more suitable for practical application environments.

4.2. Experiments. In the following, we have made a series of comprehensive experiments to simulate the execution of our scheme on a personal computer (the CPU is i7-8700U 3.2 GHz with a 24 GB memory and the operating system is Ubuntu 18.04 LTS) and the PBC library. We evaluate our scheme with different parameters and record the execution time of each step in our ASBBKS scheme. In these experiments, we assume that the user’s recursive attribute set \mathcal{A} has two levels. We show our experiment results of each step in Figure 6.

KeyGen. Figure 6(a) shows the time cost of the KeyGen phase for our ASBBKS scheme. We consider two parameters that may affect the execution time of this step: the number of subsets in a user’s set structure and the number of attributes

in each subset. We assume the number of subsets to be 1, 10, and 20 and the number of attributes in each subset to be 20, 40, 60, 80, and 100, respectively. As we can see, the time cost of KeyGen is increased linearly with the above two parameters. It only takes 638 ms when there are 20 subsets and 100 attributes in each subset. The number of attributes in each subset influences the execution time most, while the number of subsets has a smaller impact.

Encrypt. Figure 6(b) shows the time cost of the Encrypt phase. We consider three parameters that may affect the execution time of this step: the number of translating nodes and leaf nodes in an access tree and the number of keywords. We assume that there are 20, 40, 60, 80, and 100 leaf nodes and keywords and there are 0, 5, and 10 translating nodes, respectively. As we can see, the time cost of Encrypt is increased linearly with the above parameters. It only takes 990 ms when there are 10 translating nodes and 100 leaf nodes in an access tree and 100 keywords in a document. In these parameters, the main influencing factors are the number of leaf nodes and the number of keywords, while the number of translating nodes only affects the execution time for several milliseconds.

Trapdoor. Figure 6(c) shows the time cost of the Trapdoor phase. We consider three parameters that may affect the execution time of this step: the number of subsets in a user’s set structure, the number of attributes in each subset, and the number of leaf nodes in a keyword structure. We assume the number of attributes and leaf nodes to be 20, 40, 60, 80, and 100 and the number of subsets to be 1, 10, and

20, respectively. As we can see, the execution time of the Trapdoor phase is increased linearly with the above parameters. It only takes 603 ms when there are 20 subsets with 100 attributes in each subset and 100 leaf nodes for the keyword structure. In these parameters, the main influencing factors are the number of attributes and the number of leaf nodes, while the number of subsets has a smaller impact.

Test. Figure 6(d) shows the time cost of the Test phase. We consider three parameters that may affect the execution time of this step: the number of keywords, the number of attributes in each subset, and the number of translating nodes. We assume that there are 20, 40, 60, 80, and 100 attributes and keywords. We also assume that there are 0, 5, and 10 translating nodes. Further, we assume that the number of subsets in a user's set structure is fixed to 10. As we can see, the execution time of the Test phase is increased linearly with the above parameters. It only takes 370 ms when there are 100 attributes in each subset, 10 translating nodes in an access tree, and 100 keywords in the document. In these parameters, the main influencing factors are the number of attributes and the number of keywords, while the number of translating nodes has a certain impact on it.

From the above experiments for each algorithm in our scheme, we can conclude that it is an efficient and practical scheme for use in a PHR application.

5. Conclusion

In this paper, we present an attribute set-based Boolean keyword search over encrypted PHR. In our scheme, each data user's attributes are organized as recursive set structure, which enables more flexibility in user attribute organization and more efficiency in specifying policies than the existing ABKS schemes. Meanwhile, all authorized users can perform Boolean keyword search over the encrypted PHR. We have proved the security of our scheme formally. The experimental results show that it is feasible and practical for PHR systems.

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 or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was supported by the Key Areas R&D Program of Science and Technology Program of Guangzhou (202103010005) and Shenzhen Science and Technology Program (JCYJ20210324100813034).

References

- [1] K. He, J. Guo, J. Weng, J. Weng, J. K. Liu, and X. Yi, "Attribute-based hybrid boolean keyword search over outsourced encrypted data," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 6, pp. 1207–1217, 2018.
- [2] K. Kaitai, W. Liang, and W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1981–1992, 2015.
- [3] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Roşu, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries," in *Advances in Cryptology – CRYPTO 2013*, R. Canetti and J. A. Garay, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 353–373, 2013.
- [4] Y. Miao, J. Ma, X. Liu, X. Li, Q. Jiang, and J. Zhang, "Attribute-based keyword search over hierarchical data in cloud computing," *IEEE Transactions on Services Computing*, vol. 13, no. 6, pp. 985–998, 2020.
- [5] Y. Miao, X. Liu, K.-K. R. Choo et al., "Privacy-preserving attribute-based keyword search in shared multi-owner setting," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1080–1094, 2021.
- [6] L. Xu, W. Li, F. Zhang, R. Cheng, and S. Tang, "Authorized keyword searches on public key encrypted data with time controlled keyword privacy," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2096–2109, 2020.
- [7] L. Xu, X. Chen, F. Zhang et al., "ASBKS: towards attribute set based keyword search over encrypted personal health records," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 6, pp. 2941–2952, 2021.
- [8] H. Yin, J. Zhang, Y. Xiong et al., "CP-ABSE: a ciphertext-policy attribute-based searchable encryption scheme," *IEEE Access*, vol. 7, pp. 5682–5694, 2019.
- [9] U. S. Varri, S. Kumar Pasupuleti, and K. V. Kadambari, "Key-escrow free attribute-based multi-keyword search with dynamic policy update in cloud computing," in *Proceedings of the 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, pp. 450–458, IEEE, Melbourne, VIC, Australia, 11 May 2020.
- [10] H. Wang, X. Dong, and Z. Cao, "Multi-value-independent ciphertext-policy attribute based encryption with fast keyword search," *IEEE Transactions on Services Computing*, vol. 13, no. 6, pp. 1142–1151, 2020.
- [11] L. Cao, Y. Kang, Q. Wu, R. Wu, X. Guo, and T. Feng, "Searchable encryption cloud storage with dynamic data update to support efficient policy hiding," *China Communications*, vol. 17, no. 6, pp. 153–163, 2020.
- [12] S. Wang, D. Zhang, Y. Zhang, and L. Liu, "Efficiently revocable and searchable attribute-based encryption scheme for mobile cloud storage," *IEEE Access*, vol. 6, pp. 30444–30457, 2018.
- [13] D. Xiaoding Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pp. 44–55, IEEE, Berkeley, CA, USA, 14 May 2000.
- [14] E. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security. CCS '12*, pp. 965–976, Association for Computing Machinery, Raleigh, North Carolina, USA, 16 October 2012.
- [15] K. Kurosawa and Y. Ohtaki, "UC-secure searchable symmetric encryption," in *Financial Cryptography and Data*

- Security*, A. D. Keromytis, Ed., pp. 285–298, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [16] S. Kamara and C. Papamanthou, “Parallel and dynamic searchable symmetric encryption,” in *Financial Cryptography and Data Security*, A.-R. Sadeghi, Ed., pp. 258–274, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [17] G. Asharov, M. Naor, G. Segev, and I. Shahaf, “Searchable symmetric encryption: optimal locality in linear space via two-dimensional balanced allocations,” in *Proceedings Of the Forty-Eighth Annual ACM Symposium On Theory Of Computing. STOC '16*, pp. 1101–1114, Association for Computing Machinery, Cambridge, MA, USA, 2016.
- [18] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in *Advances in Cryptology - EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds., pp. 506–522, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [19] M. Abdalla, M. Bellare, D. Catalano et al., “Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions,” in *Advances in Cryptology - CRYPTO 2005*, V. Shoup, Ed., pp. 205–222, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [20] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith, “Public key encryption that allows PIR queries,” in *Advances in Cryptology - CRYPTO 2007*, A. Menezes, Ed., pp. 50–67, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [21] F. Bao, R. H. Deng, X. Ding, and Y. Yang, “Private query on encrypted data in multiuser settings,” in *Information Security Practice and Experience*, L. Chen, Yi Mu, and W. Susilo, Eds., pp. 71–85, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [22] Y. Liang, Y. Li, Q. Cao, and F. Ren, “VPAMS: verifiable and practical attributebased multi-keyword search over encrypted cloud data,” *Journal of Systems Architecture*, vol. 108, pp. 1383–7621, Article ID 101741, 2020.
- [23] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, “Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1187–1198, 2016.
- [24] Y. Miao, R. H. Deng, and X. Liu, “Multi-Authority attribute-based keyword search over encrypted cloud data,” *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 4, pp. 1667–1680, 2021.
- [25] J. Li, M. Wang, Y. Lu, Y. Zhang, and H. Wang, “ABKSSKGA: attribute-based keyword search secure against keyword guessing attack,” *Computer Standards & Interfaces*, vol. 74, pp. 0920–5489, Article ID 103471, 2021.
- [26] S. Wang, S. Jia, and Y. Zhang, “Verifiable and multi-keyword searchable attribute-based encryption scheme for cloud storage,” *IEEE Access*, vol. 7, pp. 50136–50147, 2019.
- [27] J. Shi, J. Lai, Y. Li, R. H. Deng, and J. Weng, “Authorized keyword search on encrypted data,” in *Computer Security - ESORICS 2014*, M. law Kutylowski and J. Vaidya, Eds., pp. 419–435, Springer International Publishing, Cham, 2014.
- [28] Q. Zheng, S. Xu, and G. Ateniese, “VABKS: verifiable attribute-based keyword search over outsourced encrypted data,” in *Proceedings of the IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 522–530, IEEE, Toronto, ON, Canada, 27 April 2014.
- [29] R. Bobba, H. Khurana, and M. Prabhakaran, “Attribute-sets: a practically motivated enhancement to attribute-based encryption,” in *Computer Security - ESORICS 2009*, M. Backes and P. Ning, Eds., pp. 587–604, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [30] D. Boneh and M. Franklin, “Identity-based encryption from the weil pairing,” in *Advances in Cryptology - CRYPTO 2001*, J. Kilian, Ed., pp. 213–229, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.
- [31] V. Goyal, O. Pandey, S. Amit, and W. Brent, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proceedings of the 13th ACM Conference on Computer and Communications Security. CCS '06*, pp. 89–98, Association for Computing Machinery, Alexandria, Virginia, USA, 30 October 2006.

Research Article

KLPPS: A k -Anonymous Location Privacy Protection Scheme via Dummies and Stackelberg Game

Dongdong Yang ^{1,2}, Baopeng Ye ³, Wenyin Zhang ⁴, Huiyu Zhou ⁵,
and Xiaobin Qian ⁶

¹State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China

²Guangxi Key Laboratory of Cryptography and Information Security, Guilin, China

³Information Technology Innovation Service Center of Guizhou Province, Guiyang, China

⁴School of Information Science and Engineering, Linyi University, Linyi, Shandong 276000, China

⁵School of Informatics, University of Leicester, Leicester, UK

⁶Guizhou CoVision Science and Technology Co., Ltd, Guiyang, China

Correspondence should be addressed to Wenyin Zhang; zhangwenyin@lyu.edu.cn

Received 13 October 2021; Revised 10 November 2021; Accepted 18 November 2021; Published 7 December 2021

Academic Editor: Xin-Yi Huang

Copyright © 2021 Dongdong Yang 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.

Protecting location privacy has become an irreversible trend; some problems also come such as system structures adopted by location privacy protection schemes suffer from single point of failure or the mobile device performance bottlenecks, and these schemes cannot resist single-point attacks and inference attacks and achieve a tradeoff between privacy level and service quality. To solve these problems, we propose a k -anonymous location privacy protection scheme via dummies and Stackelberg game. First, we analyze the merits and drawbacks of the existing location privacy preservation system architecture and propose a semitrusted third party-based location privacy preservation architecture. Next, taking into account both location semantic diversity, physical dispersion, and query probability, etc., we design a dummy location selection algorithm based on location semantics and physical distance, which can protect users' privacy against single-point attack. And then, we propose a location anonymous optimization method based on Stackelberg game to improve the algorithm. Specifically, we formalize the mutual optimization of user-adversary objectives by using the framework of Stackelberg game to find an optimal dummy location set. The optimal dummy location set can resist single-point attacks and inference attacks while effectively balancing service quality and location privacy. Finally, we provide exhaustive simulation evaluation for the proposed scheme compared with existing schemes in multiple aspects, and the results show that the proposed scheme can effectively resist the single-point attack and inference attack while balancing the service quality and location privacy.

1. Introduction

With the rapid development of mobile devices and social networks, location-based service (LBS) has become a vital part of our daily activities in recent years. With smartphones or tablets, users can download location-based applications from Apple Store or Google Play Store. With the help of these applications, users can easily send queries to a service provider and obtain LBSs related to some points of interest. For example, users can check the bus schedule, the price information of nearby restaurants or gas stations, etc. Undoubtedly, by

submitting LBS queries, users can enjoy the convenience provided by LBS. However, since the untrusted service provider has all the information about users such as where they are at which time, what kind of queries they submit, what they are doing, etc., he may track users in various ways or release their personal data to third parties. Thus, we need to take appropriate measures to protect users' privacy.

Many approaches [1–3] have been taken to address such privacy problems, where the location k -anonymity and location perturbation are commonly used. The existing location k -anonymity technology usually adopts the structure based

on a trusted third party (TTP) [3]. The TTP structure refers to introduce a trusted third party, called centralized location anonymizer, between the user and the service provider, and the usage of the location anonymizer to make the target user's information indistinguishable from that of at least $k - 1$ other users, so that the probability of location leakage is therefore at most $1/k$. Specifically, to achieve k -anonymity, an LBS-related query is submitted to the service provider via a centralized location anonymizer, which enlarges the queried location into a bigger cloaking region covering many other users (e.g., $k - 1$) geographically. As a result, it is hard for the untrusted service provider to distinguish the user's real location from this area. However, these approaches of using k -anonymity have a fatal problem. It heavily relies on the location anonymizer, which suffers from a single point of failure [4]. If the adversary gains control of it, the privacy of all users will be compromised.

In response to the problems existed in the TTP structure, some researchers have proposed a dummy location technology that can also achieve k -anonymity, which uses an independent system structure [2]. The independent structure only contains the user and the service provider, where the user uses the mobile terminal to generate $k - 1$ dummy locations and then sends $k - 1$ dummy locations combined with the user's real location to the service provider. As a result, it is hard for the untrusted service provider to distinguish the user's real location from the other dummy locations. Since the structure achieves functions such as location anonymity and filtering query results through a mobile terminal instead of a location anonymizer, there is no single point of failure caused by the location anonymizer. In 2008, with the birth of Bitcoin [5], blockchain technology has been widely used in finance, medical care, supply chain, and other fields. Blockchain [6–8], as the underlying technology of Bitcoin, realizes distributed information interaction and collective maintenance of data in a decentralized and autonomous way, with decentralization, tamper-proof, autonomy, traceability, etc. Simultaneously, the security of consensus protocols [9] and the protection of user privacy [10] in the blockchain has become a new research hotspot. Chen et al. [11] proposed a dynamic multikey fully homomorphic encryption. The decentralized characteristic of blockchain opens a new door for location privacy protection. Based on this idea, [12] proposes a distributed k -anonymity location privacy protection scheme based on blockchain, which can also achieve k -anonymity without the help of a location anonymizer.

In the LBS, the user firstly adopts approaches that are based on perturbing the information reported to the service provider, so to prevent the disclosure of one's location. Clearly, the perturbation of the information sent to the service provider leads to a degradation of service quality, and consequently, there is a trade-off between the level of privacy that the user wishes to guarantee and the service quality loss that she will have to accept; however, the adversary formulates corresponding strategies based on the privacy protection method adopted by the user and infers the real location of the user by observing the perturbation of the information. Since the relationship between users and adversaries objectively conforms to the game relationship between participants in the Stackelberg game model,

introducing the Stackelberg game method into the field of location privacy protection is an important research direction. Shokri et al. [13] took the lead in introducing the Stackelberg game into location privacy protection and proposed a location privacy protection scheme based on the Stackelberg game (SG). The solution assumes that the adversary has acquired prior knowledge and allows the user and the adversary to play the game in turn: The level of privacy protection is maximized by user when the service quality loss is less than a given threshold, whereas the adversary strives to minimize the level of privacy protection based on prior knowledge and offset location. By playing the game above, this strategy can finally optimize the level of privacy protection while ensuring that the service quality loss is less than a given threshold.

Based on the analysis above, the existing location privacy protection schemes have the following shortcomings: (1) the existing location privacy protection schemes either adopt TTP structure that has a single point of failure or the independent system structure. However, users in the independent structure use mobile terminals to perform location anonymity algorithms and filter query results, which will greatly increase the client's pressure, affecting the service quality in turn. (2) On the one hand, these schemes do not fully consider the semantics, physical dispersion, and query probability of the location when selecting dummy locations. On the other hand, they do not fully consider the background knowledge that the adversary may have, which adversaries can use to infer the users' location privacy information. So they cannot effectively resist single-point attacks and inference attacks. (3) Since such schemes need to sacrifice the service quality for improving the privacy protection level, there is no trade-off between service quality and privacy protection level. Aiming at related shortcomings above, this paper comprehensively considers features such as side information, location semantics, physical dispersion of locations combined with dummy locations, k -anonymity technology, Stackelberg game and other ideas, and then designs a k -anonymous location privacy protection scheme (KLPPS) based on Stackelberg game and dummy locations, which can resist single-point attacks and inference attacks while effectively balancing service quality and location privacy. Our contributions are mainly as follows:

- (1) A semitrusted third party (STTP) based location privacy protection structure is proposed. The STTP is based on the TTP structure by adding an encryption server and WiFi-AP, and stores the user's privacy information on three party servers through a certain mechanism. In the STTP, even if the adversary steals the information on the location anonymizer, he still cannot locate the user and obtain the user's complete privacy information, which effectively solves the single point of failure existed in the TTP structure. Meanwhile, the location anonymizer is responsible for implementing location anonymity algorithms and filtering query results, etc., therefore, also solve the mobile device performance bottlenecks that exist in the independent structure.

- (2) We propose a dummy location selection algorithm based on location semantics and physical distance (SPDDS). Compared with existing dummy location selection algorithms, SPDDS takes into account the characteristics such as location semantic diversity, physical dispersion, query probability and offset location when selecting dummy locations, which can effectively protect users' location privacy against single-point attack. Furthermore, we propose a location anonymous optimization method based on Stackelberg game, which introduces Stackelberg game to improve the dummy location selection algorithm. More specifically, we formalize the mutual optimization of user-adversary objectives (location privacy vs. correctness of inferring location) by using the framework of Stackelberg games, and find an optimal dummy location set by solving the game equilibrium. The optimal dummy location set can resist single-point attacks and inference attacks while effectively balancing service quality and location privacy.
- (3) We conduct a comprehensive experiment to evaluate the proposed scheme. Experimental results show that our scheme can effectively resist single-point attacks and inference attacks while effectively balancing service quality and location privacy when compared with other dummy-based schemes.

The rest of the paper is organized as follows. We discuss the related work in Section 2. Section 3 presents some preliminaries of this paper. Section 4 presents the structure of STTP and the interactive process. We present the SPDDS algorithms and a location anonymous optimization method based on Stackelberg game in Section 5. Section 6 presents the experimental process as well as results. We conclude the paper in Section 7.

2. Related Works

In this section, we first analyze the merits and drawbacks of mainstream existing location privacy protection system structure. Furthermore, we review major existing techniques for preventing location privacy leakage including privacy protection scheme based on dummy and privacy protection scheme based on Stackelberg game in Sections 2.2 to 2.3, respectively.

2.1. Location Privacy Protection System Structure. As a large body of location privacy protection technologies has been proposed, the system structure on which various privacy protection technologies depend has shown distinct category differences. As the carrier of privacy protection technology implementation, the system structure has got sufficiently researched and developed.

Currently, there are two mainstream system structures: TTP-based central server structure and independent system structure. In the TTP structure [14–16], the location anonymizer obtains the location information of all users and is responsible for implementing the location privacy

protection mechanism. It is currently a more commonly used privacy protection system structure, the advantage of which is that the location anonymizer can obtain the location information of all users and assist users in filtering service data. The disadvantage is that it relies on a location anonymizer to enlarge the queried location into a bigger cloaking region, and hence the location anonymizer becomes the central point of failure. References [2, 17, 18] proposed an independent system structure, where users can protect their location privacy according to their own capabilities and knowledge. The architecture treats each user as an independent individual, allows the user's device to implement a location privacy protection mechanism, directly sends a service request to the service provider, and receives the query result. The advantage of this system structure is that the deployment is conveniently simple, and it is convenient for users to adjust the privacy protection granularity according to their privacy protection needs. However, the implementation of privacy protection algorithms has been limited by the performance of mobile devices. Meanwhile, filtering query results will also increase the burden on the mobile client, which in turn affects service quality.

2.2. Privacy Protection Scheme Based on Dummy.

Location dummies are aimed to secure users' accurate location by sending $k - 1$ false locations ("dummies") together with the true location so that the probability of location leakage is reduced to $1/k$. Compared to the traditional k -anonymity, this approach sends exact locations instead of cloaked regions to a service provider, which can return a more precise query result and avoid single-point failure.

Kido et al. [19, 20] first proposed to use dummy locations to achieve anonymity without employing a central server. However, they only concentrate on reducing the communication costs. Moreover, they employ a random walk model to generate dummy locations, and it cannot resist side information attacks due to lack of considering factors such as query probability. Subsequently, although Dapeng et al. [21] proposed the ABR algorithm based on query probability, which can resist side information attacks. However, it cannot resist homogeneity attacks and location similarity attacks for not considering the physical dispersion and location semantic diversity. The UPHIF algorithm proposed by Chang et al. [22] protected location privacy to a certain extent, but did not consider the location semantic diversity, so it cannot deal with location similarity attacks. Niu et al. [23] selected dummy locations based on entropy metrics, and proposed a dummy location selection (DLS) scheme and its improved scheme (enhanced - DLS). Although the enhanced - DLS scheme can resist side information attacks and homogeneity attacks, which cannot resist location similarity attacks for lacking of considering the location semantic diversity. References [24, 25] all considered the user's location semantic diversity, which can effectively resist location similarity attacks, but all have the problem that the cloaked region is too big, affecting the service quality in turn. Although [26, 27] fully considered the location semantic diversity and physical dispersion, which can effectively resist

homogeneity attacks and location similarity attacks, but they cannot resist the side information attack for not considering the query probability.

2.3. Privacy Protection Scheme Based on Stackelberg Game.

In a big data environment, an adversary can use the various data collected to infer the privacy information of the user's location [28], which is called the location inference attack. Because the traditional dummy-based privacy protection scheme cannot effectively resist this kind of inference attack, the location privacy protection mechanism based on probabilistic reasoning [29, 30] has gradually attracted the attention of scholars. Such methods are based on perturbing the real locations of a user to the service provider, in order to increase the uncertainty of the adversary about a user's true whereabouts. However, the perturbation of the information sent to the service provider leads to a degradation of service quality, and consequently, there is a trade-off between the level of privacy that the user wishes to guarantee and the service quality loss that she will have to accept. So, the Stackelberg game has become an important means of balancing the level of privacy protection and service-quality requirements in such methods.

Based on [13] and combined the ideas of k -anonymity and dummy location, Xingyou et al. [31] propose HCLS and its improved scheme (HCLS – SG). Although the HCLS – SG scheme can effectively resist inference attacks and better balance service quality and location privacy, it cannot resist location similarity attacks for not considering location semantic diversity. Bordenabe et al. [32] also introduced differential privacy on the basis of [13] and constructed a privacy protection mechanism that optimizes the quality of service. Since differential privacy does not depend on prior, this mechanism can minimize the service quality loss under the premise of satisfying location indistinguishability. Shokri [18] further proposed using two indicators of differential privacy and distortion privacy to optimize the privacy protection strategy based on the Stackelberg game. Differential privacy limits the extent of user privacy leakage, while distortion privacy measures the error rate of inferring a user's privacy. By combining these two standards, this privacy protection strategy can resist more kinds of inference attacks while ensuring privacy protection requirements.

3. Preliminaries

In this section, we first introduce some relative definitions of location privacy protection algorithm; meanwhile, we summarize the notations introduced throughout the section in Table 1 and then introduce the relative concepts of Stackelberg game.

3.1. Relative Definitions of Location Privacy Protection Algorithm

Definition 1. According to [27], location semantic tree (LST), a true structure used to represent the semantic

relations between two locations within the range of a Wi-Fi access points (Wi-Fi AP), which satisfies the following requirements:

- (a) Each nonleaf node stands for the category of its children nodes and each leaf node for a real location l
- (b) The depth of LST, denoted as h , is equal to the maximum number of layers of categories plus 1
- (c) The semantic distance $d_{\text{sem}}(l_i, l_j)$ between two locations $l_i, l_j (i \neq j)$ is the number of hops from leaf node n_i to leaf node n_j

Definition 2. User's privacy requirements S , represented by two-tuple (k, u) that has the following meanings:

- (a) k denotes the anonymous degree of our location privacy preservation model. More specifically, each query is sent with at least $k - 1$ dummy locations and its offset location (we use offset location instead of the real location), making that the probability of offset location leakage is therefore $1/k$.
- (b) u represents the minimum acceptable value of semantic distance between two locations in dummy location set (DLS). In other words, it satisfies the inequality:

$$\min [d_{\text{sem}}(l_i, l_j)] \geq u. \quad (1)$$

Definition 3 (location map distance). If we let Map_{cur} represent the map information within the range of the current Wi-Fi AP. For any two locations $l_i, l_j (i \neq j)$, the location map distance is the physical distance between the two locations on Map_{cur} , the value of which ranges from tens of meters to hundreds.

Definition 4 (location query probability (LQP)). As shown in Figure 1, in a map divided into $m \times m$ cells with equal size. Each cell has a query probability based on the previous query history, which is denoted as

$$p_i = \frac{\text{number of queries in cells}}{\text{number of queries in whole map}}, \quad (2)$$

where $i = 1, 2, \dots, m^2, \sum_{i=1}^{m^2} p_i = 1$. The depth of the color in the figure indicates LQP (the darker the color, the greater the LQP), and the white area indicates that the location has never had a location service request, so these locations may be rivers, barren mountains, and other places that are easily filtered by the adversary.

Definition 5. The probability of exposing real location (PERL), which has been used to measure the effectiveness of the algorithm against side information attacks, is computed by

$$\text{PERL} = \frac{1}{k - k'}, \quad (3)$$

where k denotes the anonymous degree and k' represents the number of dummy locations filtered by the adversary

TABLE 1: Summary of notations.

Symbol	Meaning
$d_{\text{sem}}(l_i, l_j)$	The semantic distance between two locations $l_i, l_j (i \neq j)$
$d_{\text{phy}}(l_i, l_j)$	The physical distance between two locations $l_i, l_j (i \neq j)$
$d_{\text{que}}(l_i, l_j)$	The query probability distance between two locations $l_i, l_j (i \neq j)$, which is obtained by calculating the difference between the query probabilities of two locations
θ	Representation of the semantic diversity between locations
$\varphi(l)$	Location access profile of the user (probability of being at location when accessing the LBS)
l	Actual location l of the user
l_d	Offset location output by the $f(l_d l)$
DLS	Set of possible dummy locations output by the LPPM
$f(\text{DLS} l)$	The location privacy protection mechanism (LPPM): probability of replacing l with DLS
\hat{l}	Adversary's estimate of the user's actual location
$g(\hat{l} \text{DLS})$	Adversary's attack function: probability of estimating \hat{l} as user's actual location, if DLS is observed
$f(l_d l)$	Function of generating offset location: probability of replacing l with l_d
Q_{loss}	Expected quality loss of an LPPM with location obfuscation function $f(l_d l)$
$Q_{\text{loss}}^{\text{max}}$	Maximum tolerable service quality loss
PL	Expected location privacy of the user with profile $\varphi(l)$ using LPPM $f(\text{DLS} l)$ against attack $g(\hat{l} \text{DLS})$

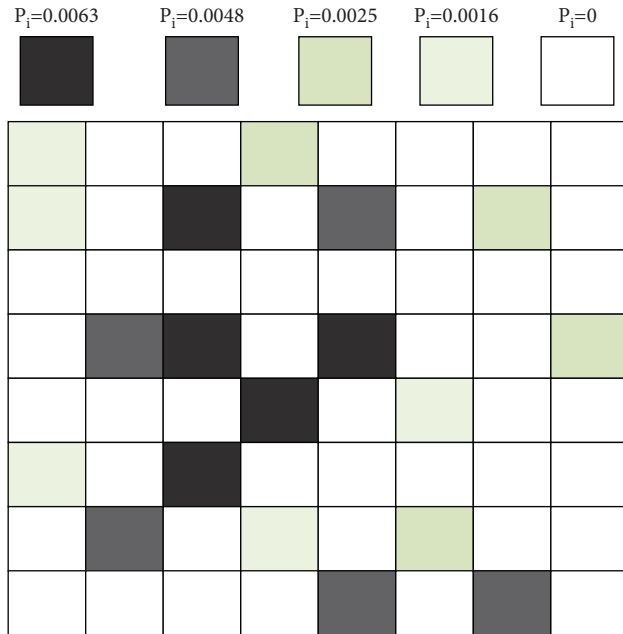


FIGURE 1: Location query probability diagram.

through the side information attack. The larger the PERL, the less effective the algorithm resists side information attacks; the smaller the PERL, the better the algorithm resists side information attacks.

Definition 6. Location physical dispersion (PD), which has been used to measure the effectiveness of the algorithm against location homogeneity attacks, is obtained by computing the minimum physical distance between any two locations in a DLS. The specific process is shown in (4):

$$\text{PD} = \text{Min}[d_{\text{phy}}(l_i, l_j)], \quad (4)$$

where $i, j = 1, 2, \dots, k, i \neq j$. The greater the minimum distance between any two locations in the DLS, the greater the PD and the greater the coverage of the DLS, the better the algorithm's resistance to location homogeneity attacks.

Definition 7. θ -secure set of dummy locations. Dummy location set (DLS) consisting of $k - 1$ dummy locations and the offset location, where the semantic distance between l_i and l_j satisfies

$$1 - \frac{|\text{SEM}|}{C_k^2} \geq \theta, \quad (5)$$

where $\text{SEM} = \{d_{\text{sem}}|d_{\text{sem}}(l_i, l_j) < u\}$, $k = |\text{DLS}|$, and C_k^2 is a combination formulas, we call DLS a θ -secure set. We use θ as a privacy protection index of location semantics in our experimental analysis in Section 6. Our aim is to achieve the maximum θ , i.e., to make it equal to 1, such that two locations in DLS belong to different categories.

Definition 8. The adversary uses background knowledge to run an inference attack on DLS in order to output estimation \hat{l} of the user's actual locations and the attack result can be denoted as $g(\hat{l}|\text{DLS})$, and we define the location privacy protection mechanism (LPPM) that the adversary knows as $f(\text{DLS}|l)$. Then, we follow the definition in [33] and quantify the user's privacy level (PL) as the adversary's expected error in his inference attack, i.e., the expected distortion in the reconstructed event. We compute the expectation over all l, \hat{l} , and DLS:

$$\text{PL} = \sum_{l, \hat{l}, \text{DLS}} \varphi(l) f(\text{DLS}|l) g(\hat{l}|\text{DLS}) d_{\text{phy}}(l, \hat{l}), \quad (6)$$

PL directly reflects the adversary's inference on the user's actual location. The larger the PL, the less accurate the adversary's inference, and the better the effect of the algorithm resist the inference attack.

Definition 9. We define the process of generating the offset location l_d as $f(l_d|l)$. Then, following the definition in [13], the LBS response quality depends on the offset location l_d output by $f(l_d|l)$ and not on the user's actual location l . The distortion introduced in the offset location determines the quality of service that the user experiences. The more similar the actual and the offset location are, the higher the service

quality is. The expected quality loss (Qos_{loss}) due to $f(l_d|l)$ is computed as an average of $d_{\text{phy}}(l, l_d)$ over all l and l_d :

$$Qos_{\text{loss}} = \sum_{l, l_d} \varphi(l) f(l_d|l) d_{\text{phy}}(l, l_d). \quad (7)$$

We set a service quality threshold $Q_{\text{loss}}^{\text{max}}$, which represents the maximum service quality loss that the user can accept. The location privacy protection scheme designed in this paper needs to guarantee $Qos_{\text{loss}} \leq Q_{\text{loss}}^{\text{max}}$ because if the service quality loss exceeds the threshold, the location service request results obtained cannot satisfy the requirements of users.

3.2. Stackelberg Game. The classic Stackelberg game is a two-player game composed of a leader and a follower [34]. The leader first determines his strategy, and after observing the leader's strategy, the follower chooses the strategy that maximizes his utility to play the game. In the field of location privacy protection, the terms "protector" and "leader," "adversary" and "follower" can be used interchangeably. For simplicity of expression, the leader (protector), denoted as Θ , is often referred to as she, whereas the follower (adversary), represented by Ψ , is referred to as he.

Definition 10. In the field of location privacy protection, the strong Stackelberg equilibrium (SSE) is generally used as the solution of the Stackelberg game. The definition of SSE is described as follows.

A strategy combination is SSE, if and only if it satisfies the following conditions:

- (a) The leader's strategy is the best response:

$$\sum_{\gamma \in \Gamma} \Omega_{\Theta}(x, q, \gamma) \geq \sum_{\gamma \in \Gamma} \Omega_{\Theta}(x', q, \gamma) \quad \forall x, x' \in X, q \in Q, \quad (8)$$

where $\gamma \in \Gamma$ means a particular type of follower, $x, x' \in X$ represents the leaders' mixed strategy, and $q \in Q$ represents the followers' mixed strategy.

- (b) The follower's strategy is the best response:

$$\sum_{\gamma \in \Gamma} \Omega_{\Psi}(x, q, \gamma) \geq \sum_{\gamma \in \Gamma} \Omega_{\Psi}(x, q', \gamma) \quad \forall x \in X, q, q' \in Q, \quad (9)$$

where $x \in X$ represents the leaders' mixed strategy, $q, q' \in Q$ represents the followers' mixed strategy.

- (c) If there are multiple best responses for the followers, the followers choose the most favorable strategy for the leader to break the deadlock:

$$\sum_{\gamma \in \Gamma} \Omega_{\Psi}(x, q', \gamma) \geq \sum_{\gamma \in \Gamma} \Omega_{\Psi}(x, q'', \gamma) \quad \forall x \in X, q', q'' \in Q^*(x), \quad (10)$$

where $Q^*(x)$ is the follower's best response strategy set under the leader's strategy is x .

4. System Model

We first give the definition of the single-point attack mode and inference attack mode in Sections 4.1 and 4.2 respectively, and then introduce the structure of STTP in Section 4.3. Finally we present the interactive process of our scheme in Section 4.4.

4.1. Single-Point Attack Model. From the time dimension, the adversary relies on the intercepted single location-service request to infer the user's private information, which is called the single-point attack model [35]. In the model, the main attack methods of adversaries include side information attacks, homogeneity attacks and location similarity attacks.

Side information refers to information used by adversaries to filter dummy locations and help reduce anonymity, including map information and location query probability. For example, for a randomly generated dummy location set, some locations may be in a river or no man's land, and adversaries can easily filter out these locations based on the map information. Assuming that the location anonymity requirement is k , when k' of the $k - 1$ dummy locations are filtered by the adversary based on the side information, the k -anonymity requirement is not satisfied, resulting in a decrease in the level of privacy protection.

Homogeneity attack means that the adversary analyzes the distance between multiple locations in a DLS to infer a user's privacy. Specifically, if the distance between any two locations is very close such as in the same building, although the DLS satisfies k -anonymity, the user's location privacy cannot be well protected because the cloaking region is too small.

The location similarity attack means that the adversary analyzes the semantic information in the cloaking region to infer a user's privacy. More specifically, if the region contains only one kind of semantic information, such as a hospital or school, the adversary can infer the user's behavior.

4.2. Inference Attack Model. In a big data environment, an adversary can use the various data collected to infer the privacy information of the user's location [28], which is called the location inference attack.

In the location inference model, the adversary has certain background knowledge such as the user's service request history records, LPPM, etc. Using the user's service request history records, the adversary can calculate the user's query probability distribution $\varphi(l)$. When the user sends a query request again, if the location query probability distribution in the anonymous set is not uniform, the adversary can infer that the user is likely to be located in a location with a higher probability. While for the LPPM, the adversary can analyze the intercepted location request, combined with the anonymity algorithm, to infer the probability that each location in the anonymous set is the user's true location, so as to make the inference attack more accurate.

4.3. The Structure of STTP. It can be seen from Section 2.1 that, for the current two mainstream location privacy protection system structures, the TTP structure has the

problem of a single point of failure, while the independent structure has the problem of mobile device performance bottlenecks. In view of the problems above, we have designed a semi-trusted third party (STTP) based location privacy protection structure. STTP is based on the traditional TTP structure by adding an encryption server and Wi-Fi AP and stores the user's private information in the three-party server through a certain mechanism, which results in that even if the location anonymizer has been controlled by adversaries, STTP also protects the user's private information to a certain extent. Furthermore, the location anonymizer is responsible for implementing the privacy protection algorithms and filtering query results, so there are no problems such as mobile device performance bottlenecks. STTP is shown in Figure 2, which consists of the following five entities:

User: using a mobile terminal to initiate a location service request when needed.

Wi-Fi AP: providing network support, and calculating, storing LST and LQP.

Encryption server (ES): providing encryption and decryption key pairs corresponding to the user's pseudonym.

Location anonymizer (LA): converting the user's actual location into a dummy location set, and after the service provider returns the query result, extracting appropriate service information and returning it to the user.

Service provider (SP): return the corresponding service result according to the location query request.

The proposed scheme assumes that ES, LA, and SP are "honest and curious." On the one hand, they will not disrupt the protocol process and can faithfully complete their work following the agreement; on the other hand, they all want to analyze more other sensitive information about the user from what they have mastered. Meanwhile, we further set that ES, LA, and SP cannot collude with each other, that is, they will not be controlled by an adversary simultaneously. There will be no secrets for the user if the three parties conspire, so this setting is reasonable.

4.4. Interactive Process. There are eight steps in the interactive process of the proposed scheme. The specific implementation of each step is described below (as shown in Figure 2):

- (1) Before initiating a location service request, the user first requests Map_{cur} , LST, and LQP from the Wi-Fi AP.
- (2) The Wi-Fi AP generates Map_{cur} , computes and stores LQP of all locations within its current coverage area, generates and saves LST by collecting location semantic information within its radio range, and then sends Map_{cur} , LST, and LQP to the user. It should be noted that for any Wi-Fi AP, the location within its

coverage area is relatively stable, so LST and LQP do not need to change frequently.

- (3) The user then requests the pseudonym U_{pseu} and key pair from ES. Specifically, if there are multiple service requests at the same location within the limited time t_{session} , the user only applies for the pseudonym and key pair once; when the time exceeds t_{session} or the user's real location changes, she will reapply for a new pseudonym and key, so as to achieve the effect of confusing her identity.
- (4) ES generates the corresponding pseudonym U_{pseu} and RSA key pair $(\text{Key}_{\text{public}}, \text{Key}_{\text{privacy}})$, returns U_{pseu} and $\text{Key}_{\text{public}}$ to the user, and sends U_{pseu} and $\text{Key}_{\text{privacy}}$ to the SP. It should be noted that, as an example, the solution uses the classic RSA algorithm for encryption, and it can be replaced by other encryption algorithms according to actual requirements. In addition, the solution requires ES to only act as a provider of pseudonyms and keys, so ES does not store related pseudonyms and keys locally.
- (5) The user first encrypts his query content Query with the public key $\text{Key}_{\text{public}}$ and then sends his current pseudonym U_{pseu} , encrypted query content Query' , current real location l , Map_{cur} , LQP, and LST to LA together.
- (6) After receiving the information, LA performs the corresponding location anonymity algorithm that generates a dummy location set DLS to hide l and then sends U_{pseu} , Query' and DLS to the SP.
- (7) After receiving the location service request, the SP first searches for the corresponding private key $\text{Key}_{\text{privacy}}$ according to U_{pseu} , which is used to decrypt Query' , and then provides the service result $\text{Result}(\text{Query}|\text{DLS})$ according to Query and DLS, finally return it to LA.
- (8) After receiving $\text{Result}(\text{Query}|\text{DLS})$, the LA first identifies the corresponding location l_d according to the U_{pseu} , and then filters out the query result $\text{Result}(l_d)$ from $\text{Result}(\text{Query}|\text{DLS})$ and finally returns it to the user.

5. Proposed Scheme

In this section, we first introduce a dummy location selection algorithm based on location semantics and physical distance (SPDDS) and then present a location anonymous optimization method based on Stackelberg game.

5.1. A Dummy Location Selection Algorithm Based on Location Semantics and Physical Distance. Based on the analysis above, the final dummy location set not only needs to avoid selecting places that are easy to be filtered by adversaries, such as rivers and no man's land but also to meet the semantic diversity while making the locations as dispersed as possible. In other words, the final dummy location set needs to simultaneously satisfy (11)–(13)

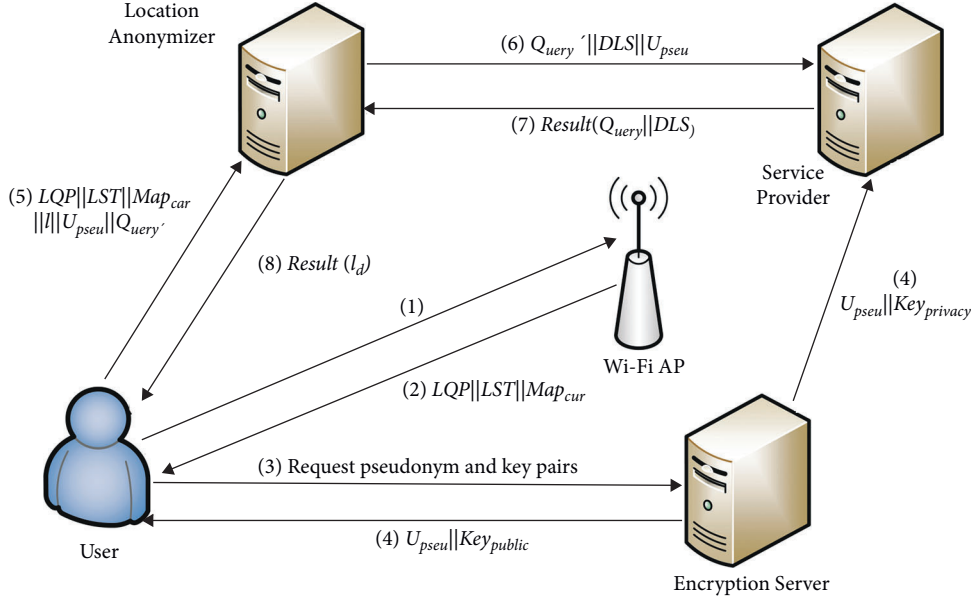


FIGURE 2: A semitrusted third party based location privacy protection structure.

$$DLS = \arg \min \{ \max [d_{\text{que}}(l_i, l_j)] \}, \quad (11)$$

$$DLS = \arg \max \{ \min [d_{\text{sem}}(l_i, l_j)] \}, \quad (12)$$

$$DLS = \arg \max \{ \min [d_{\text{phy}}(l_i, l_j)] \}, \quad (13)$$

where $l_i, l_j \in DLS, i \neq j, P_r(l_i) > 0, P_r(l_d) > 0$. It can be formulated as a multiobjective optimization problem (MOP) since three factors are considered simultaneously. However, we put forward a simpler objective formulas considering the complexity of MOP. In each dummy location set, we would like to make sure that (14) can be satisfied. Consequently, we propose a dummy location selection algorithm based on location semantics and physical distance (SPDDS).

$$DLS = \arg \max \left\{ \frac{\min [d_{\text{sem}}(l_i, l_j) + r \cdot d_{\text{phy}}(l_i, l_j)]}{\max [d_{\text{que}}(l_i, l_j) + 1]} \right\}, \quad (14)$$

where $l_i, l_j \in DLS, i \neq j$. $d_{\text{que}}(l_i, l_j) + 1$ is to avoid the situation where the two locations have the same probability, that is, the difference between the query probability of the two locations is 0. Here, r is a controllable factor for balancing the share of semantic distance, physical distance, and query probability distance since $d_{\text{sem}}(l_i, l_j) \leq 2 \cdot (h - 1)$, where h is the depth of LST and hence is usually less than 10 while $d_{\text{phy}}(l_i, l_j)$, as Wi-Fi transmission distance, ranges from hundreds of meters to thousands, whereas the query probability distance is always less than 1. Consequently, we set $r = 0.03$.

Meanwhile, in order to balance service quality while the proposed algorithm can resist inference attacks effectively, we should take into account both PL and QoS_{loss} . So, we propose a location anonymous optimization method based on Stackelberg game, which introduces Stackelberg game to optimize the dummy location selection algorithm. More specifically, we

formalize the mutual optimization of user-adversary objectives (location privacy vs. correctness of inferring location) by using the framework of Stackelberg games, to find an optimal dummy location set. The optimal dummy location set can resist single-point attacks and inference attacks while effectively balancing service quality and location privacy.

The main idea of SPDDS is to first select an offset location to replace the user's real location; secondly, selecting all locations that satisfy the semantic difference with the existing locations in the current dummy location set as the dummy location candidate set (DLCS); then, selecting an optimal location in the DLCS which refers to the location formed by satisfying (14); finally, a set consisting of an offset location and $k - 1$ dummy locations is generated. Algorithm 1 shows the formal description of the SPDDS algorithm.

5.2. A Location Anonymous Optimization Method Based on Stackelberg Game. We propose a location anonymous optimization method based on Stackelberg game, which optimizes the dummy location selection algorithm by introducing the Stackelberg game. More specifically, we formalize the mutual optimization of user-adversary objectives (location privacy vs. correctness of inferring location) by using the framework of Stackelberg game, and based on which we construct the related linear programs. We can find an optimal dummy location set by solving the linear programs, which can resist single-point attacks and inference attacks while effectively balancing service quality and location privacy.

5.2.1. Location Inference Model. In the location inference model, the adversary has certain background knowledge such as the user's service request history records, LPPM, etc. Using the user's service request history records, the adversary can calculate the user's query probability distribution $\varphi(l)$. When the user sends a query request again, if the

Input: l : user's real location

S : user's privacy requirement

Map_{cur} : map information in current Wi-Fi AP LST: location semantic tree LQP: location query probability;

Output: DLS: dummy location set;

- (1) Divide the Map_{cur} as the sample space into $m \times m$ grids;
- (2) Generate semantic distance matrix (S DM) according to the LST and geographic distance matrix (GDM) according to the Map_{cur} and probability distance matrix (PDM) according to the LQP respectively
- (3) According to GDM and PDM, choose the $n - 1$ locations with $p_i > 0$, which are closest to l , and then a offset location set (M_{set}) consisting of the $n - 1$ locations and l has been generated;
- (4) Randomly choose a location from M_{set} as offset location l_d
- (5) Generate a dummy location candidate set (DLCS) for all $p_i > 0$ locations in the whole grid space;
- (6) $\text{DLS} = \{l_d\}$
- (7) Remove l_d from DLCS;
- (8) **while** $|\text{DLS}| < k$ **do**
- (9) **if** DLCS = ϕ **then**
- (10) anonymity failed;
- (11) **else**
- (12) $\text{max} = 0$; $\text{BestLoc} = \phi$;
- (13) **for each** Loc in DLCS **do**
- (14) **if** $d_{\text{sem}}(\text{Loc}, \text{DLS.last}) \leq u$ **then**
- (15) anonymity failed;
- (16) go back to line 13;
- (17) **else**
- (18) $m = d_{\text{sem}}(\text{Loc}, \text{DLS.last}) + r \cdot [d_{\text{phy}}(\text{Loc}, \text{DLS.last})/d_{\text{que}}(\text{Loc}, \text{DLS.last})]$
- (19) compute the maximum m according to SDM, GDM and PDM, which is recorded with max , and then assign the corresponding Loc to BestLoc
- (20) **end**
- (21) **end**
- (22) $\text{DLS} = \text{DLS} \cup \{\text{BestLoc}\}$
- (23) remove BestLoc from DLCS;
- (24) **end**
- (25) **end**
- (26) output DLS;

ALGORITHM 1: A dummy location selection algorithm based on location semantic and physical distance.

location query probability distribution in the anonymous set is not uniform, the adversary can infer that the user is likely to be located in a location with a higher probability. While for the LPPM, the adversary can analyze the intercepted location request, combined with the anonymity algorithm, to infer the probability that each location in the anonymous set is the user's true location, so as to make the inference attack more accurate.

Based on the existing knowledge ($\varphi(l)$, $f(\text{DLS}|l)$, etc.), the adversary can form the posterior distribution on the true location l of the user, conditional on the anonymous set DLS:

$$P_r(l|\text{DLS}) = \frac{P_r(l, \text{DLS})}{P_r(\text{DLS})} = \frac{\varphi(l)f(\text{DLS}|l)}{\sum_l \varphi(l)f(\text{DLS}|l)}. \quad (15)$$

The adversary's objective is then to choose \hat{l} to minimize the user's conditional expected privacy, where the expectation is taken under $P_r(l|\text{DLS})$. The user's conditional expected privacy for an arbitrary \hat{l} is

$$\sum_l P_r(l|\text{DLS})d_{\text{phy}}(l, \hat{l}), \quad (16)$$

and for the minimizing \hat{l} , it is

$$\min_{\hat{l}} \sum_l P_r(l|\text{DLS})d_{\text{phy}}(l, \hat{l}). \quad (17)$$

If there are multiple values of \hat{l} that satisfy (17), then the adversary randomizes arbitrarily among them. The probability with which \hat{l} is chosen in this randomization is $g(\hat{l}|\text{DLS})$.

5.2.2. Stackelberg Game Optimization Process. Here, we assume that the adversary has some background knowledge. Specifically, he will infer the user's actual location l as much as possible according to $\varphi(l)$, the LPPM used by the LA, the anonymous result DLS, and other background knowledge. Relatively, we can assume that all the background knowledge that LA knows will be used by the adversary, so LA can use the adversary's optimal attack strategy as a parameter to optimize the generation process of the dummy location set DLS.

We formalize the process above by using the framework of Stackelberg game. In a Stackelberg game the leader, in our case, the LA plays first by giving the dummy location set DLS according to the relative location privacy protection

algorithm $f(DLS|l)$. The follower, in our case the adversary, plays next by estimating the user's true location, knowing some background knowledge.

We use the distance between the adversary's inferred location \hat{l} and the user's actual location l to measure the utility of the participants in the game: the greater the distance, the greater the LA returns, indicating that the anonymous algorithm is more effective in resisting inference attacks; on the contrary, the smaller the distance, the greater the adversary returns, the more effective the adversary's attack strategy.

The game model is also an instance of a zero-sum game, as the adversary's gains (or losses) of utility is exactly balanced by the losses (or gains) of the utility of the user: the information gained (lost) by the adversary is the location privacy lost (gained) by the user.

The purpose of Stackelberg game optimization is to find SSE so that the adversary cannot obtain more benefits by optimizing the attack strategy (that is, the adversary cannot make more accurate inferences about the actual location of the user). In this paper, SPDDS optimized by Stackelberg game is denoted as SPDDS_SG.

It should be noted that DLS is the result obtained by l_d further anonymously, so $f(DLS|l)$ can be further expressed by

$$f(DLS|l) = f(l_d|l) \cdot f(DLS|l_d). \quad (18)$$

In some cases, $f(DLS|l)$ and $f(l_d|l)$ are equal, the reason is that the adversary can filter out dummy locations except l_d in such cases.

We see that, for a given DLS, the user's conditional expected privacy is given by (17). The probability that DLS is output by the LPPM is $P_r(DLS) = \sum_l \varphi(l) f(DLS|l)$. Hence, the user's unconditional expected privacy (averaged over all DLS) is

$$\sum_{DLS} P_r(DLS) \min_l \sum_l P_r(l|DLS) d_{\text{phy}}(l, \hat{l}) = \sum_{DLS} \min_l \sum_l \varphi(l) f(DLS|l) d_{\text{phy}}(l, \hat{l}). \quad (19)$$

To facilitate the computations, we define

$$x \triangleq \min_l \sum_l \varphi(l) f(DLS|l) d_{\text{phy}}(l, \hat{l}). \quad (20)$$

Incorporating x into (19), we write the unconditional expected privacy of the user as

$$\sum_{DLS} x, \quad (21)$$

which the user aims to maximize by choosing the optimal DLS. To facilitate the computations, (20) can be transformed to a series of linear constraints:

$$x \leq \sum_l \varphi(l) f(DLS|l) d_{\text{phy}}(l, \hat{l}) = \sum_l \varphi(l) f(l_d|l) f(DLS|l_d) \cdot d_{\text{phy}}(l, \hat{l}), \forall \hat{l}. \quad (22)$$

In addition, SPDDS_SG needs to conceal the user's real location on the premise of ensuring the user's service quality. In order to ensure the quality of service, we set the service

quality threshold $Q_{\text{loss}}^{\text{max}}$ to limit the maximum service quality loss. The specific process is

$$\sum_{l, l_d} \varphi(l) f(l_d|l) d_{\text{phy}}(l, l_d) \leq Q_{\text{loss}}^{\text{max}}. \quad (23)$$

In summary, SPDDS_SG can be solved by a linear program. The final definition of linear program is

$$\begin{aligned} & \text{Maximize } \sum_{DLS} x, \\ & \text{s.t. } C_1: x \leq \sum_l \varphi(l) f(l_d|l) f(DLS|l_d) d_{\text{phy}}(l, \hat{l}), \\ & C_2: \sum_{l, l_d} \varphi(l) f(l_d|l) d_{\text{phy}}(l, l_d) \leq Q_{\text{loss}}^{\text{max}}, \\ & C_3: \sum_{DLS} f(DLS|l) = 1, \\ & C_4: f(DLS|l) \geq 0, \forall l, DLS, \end{aligned} \quad (24)$$

where C_1 is used to maximize the utility of the adversary; C_2 reflects the service quality constraint; C_3 indicates that the sum of the generation probability of the dummy location set must be 1; C_4 indicates the probability of each candidate dummy location set is greater than zero.

SPDDS_SG solves the objective function under the constraints in (24) and obtains the optimal dummy location set, which can resist single-point attacks and inference attacks while effectively balancing service quality and location privacy.

6. Simulations and Results

In this section, we use Python software to simulate the experiment. First, we give the relevant parameters of the experiment. Furthermore, we simulate the experimental results and analysis of the proposed scheme.

6.1. Simulation Setup. Our scheme is implemented in MATLAB and performed on a Windows 10 PC with an Intel Core i5-8500 CPU, a 3.00 GHz processor and a 8.00 GB main memory. We use a real road map of Guangzhou from Google Maps, since Guangzhou as a provincial capital in southern China is a big city with enough users in LBS and its central urban area has been covered by Wi-Fi APs in 2016. The coverage area of each Wi-Fi AP is about 700 ~ 800 m, the sample space is divided into 13×13 cells with equal size, and a total of 13 559 sample trajectories are used as historical data to calculate the historical query probability of each cell. Besides, all locations in our experiments are divided into 6 categories semantically as follows: Education and Science, Administration and Housing, Medical care, Shopping malls, Public places, Catering and Entertainment. The value ranges of the main parameters u and k of the experiment are $3 \leq u \leq 7$ and $2 \leq k \leq 30$, respectively.

6.2. Results and Analysis. We first evaluate the effectiveness of our proposed scheme in resisting single-point attacks from three assessment metrics as follows: (1) PERL. As is shown in

Definition 5, it reflects the effectiveness of the algorithm in resisting side information attack. (2) PD. As is shown in Definition 6, the larger the PD, the more dispersed the dummy locations in the DLS, and the better the effectiveness of the algorithm in resisting homogeneity attack. (3) θ . As is shown in Definition 7, it refers to the level of semantic diversity in the anonymous set, which reflects the effectiveness of the algorithm in resisting location similarity attack; Next, evaluating the effectiveness of the scheme in resisting inference attack while balancing location privacy and service quality from two assessment metrics as follows: (1) PL. As is shown in Definition 8, the larger the PL is, the better the effect of LPPM against inference attacks is. (2) QoS_{loss} . As is shown in Definition 9, it reflects the effectiveness of the algorithm in balancing location privacy and service quality.

6.2.1. Effectiveness of the Scheme against Single-point Attacks.

(1) *PERL vs k*. In Figure 3(a), we compare the PERL of KLPPS, Max Min Dist DS [27], Simp Max Min Dist DS [27], enhanced – DLS [23], and HCLS – SG [31] schemes. As we can see, the PERL of the five schemes shows a downward trend with the increase of k , which means that the larger the k , the more difficult it is for adversaries to filter out invalid locations in the anonymous set through side information attacks, the better the effect of the scheme against side information attacks. The PERL of the KLPPS, enhanced – DLS, and HCLS – SG is lower than that of the Max Min Dist DS and Simp Max Min Dist DS. And that of the KLPPS, enhanced – DLS, and HCLS – SG are basically the same. The reason is that the KLPPS, enhanced – DLS and HCLS – SG all consider the query probability and avoid selecting locations with low access probability such as lakes and forests to form an anonymous set; whereas the Max Min Dist DS and Simp Max Min Dist DS do not consider the query probability, so there will be cases where invalid locations are selected, and thereby the adversary can filter out ones through side information attacks. In summary, the KLPPS scheme can effectively resist side information attacks.

(2) *P D vs k*. Figure 3(b) shows the PD comparison chart of KLPPS, Max Min Dist DS, Simp Max Min Dist DS, enhanced – DLS, and HCLS – SG schemes. As we can see, the PD of KLPPS, HCLS – SG, enhanced – DLS, and Max Min Dist DS are close when $k \leq 4$; at $k \geq 5$, the PD of Max Min Dist DS is slightly larger than that of KLPPS, HCLS – SG and enhanced – DLS. Under the same value of k , the PD of KLPPS, HCLS – SG, and enhanced – DLS is slightly larger than that of Simp Max Min Dist DS. In additional, with the increase of k , the PD of the five schemes are both reduced gradually. The reason for this is obvious: it becomes harder to maintain a high level of dispersion with more and more dummies. In summary, Max Min Dist DS has the largest PD, KLPPS, HCLS – SG, enhanced – DLS, and Simp Max Min Dist DS decrease in order, which means that the Max Min Dist DS is better in resisting homogeneity attacks than the other four schemes, but the KLPPS scheme is also acceptable.

(3) *θ vs k*. Figure 3(c) shows the value of θ comparison between KLPPS, Max Min Dist DS, Simp Max Min Dist DS, enhanced – DLS, and HCLS – SG schemes. As shown in Figure 3(c), with the increases of k , the value of θ of KLPPS, Max Min Dist DS, and Simp Max Min Dist DS schemes hardly change and close to the maximum value 1. However, that of enhanced – DLS and HCLS – SG schemes is always at a relative low. The reason is that the KLPPS, Max Min Dist DS, and Simp Max Min Dist DS schemes all consider the semantic information of the location when selecting dummy locations, thereby ensuring semantic diversity, while the enhanced – DLS and HCLS – SG schemes only consider the query probability of each location point instead of considering the situation that each location point may have the same semantic information. Moreover, the location points with higher query probability are often in hotspot areas, between which the semantic information is very similar and therefore not satisfying the semantic diversity. Consequently, the enhanced – DLS and HCLS – SG schemes behave such badly in semantic diversity that they cannot resist location similarity attacks. In summary, the KLPPS scheme can effectively resist location similarity attacks.

The experimental results above show that the KLPPS scheme can effectively resist homogeneity attacks, location similarity attacks, and side information attacks simultaneously compared with the Max Min Dist DS, Simp Max Min Dist DS, enhanced – DLS, and HCLS – SG schemes, thereby effectively resisting single-point attacks.

6.2.2. Effectiveness of the Scheme against Inference Attacks and Balances PL and QoS_{loss} .

Combining the location inference model and (6), it can be seen that the adversary can perform inference attacks, the purpose of which is to choose \hat{l} based on existing knowledge to minimize the expected user privacy. (25) defines this attack strategy:

$$\hat{l} = \underset{\hat{l}}{\operatorname{argmin}} \text{PL}. \quad (25)$$

Combining (6) and (25), we can construct the following linear program to find the optimal \hat{l} :

$$\begin{aligned} & \underset{\hat{l}, \text{DLS}}{\operatorname{minimize}} \sum \varphi(l) f(\text{DLS}|l) g(\hat{l}|\text{DLS}) d_{\text{phy}}(l, \hat{l}), \\ & \text{s.t. } C_1: \sum_{\hat{l}} g(\hat{l}|\text{DLS}) = 1, \forall \text{DLS}, \\ & C_2: g(\hat{l}|\text{DLS}) \geq 0, \forall \text{DLS}, \hat{l}. \end{aligned} \quad (26)$$

We use the model defined by (26) to run inference attacks on KLPPS, HCLS [31], SG [13], and HCLS – SG to make a comparison from two aspects of PL and QoS_{loss} , evaluating the effectiveness of the KLPPS scheme.

(1) *PL*. The definition of PL is shown in (6), the larger the PL, the better the effect of LPPM against inference attacks. As shown in (24), the preset service quality loss threshold $Q_{\text{loss}}^{\text{max}}$ and anonymous degree k have a greater impact on PL,

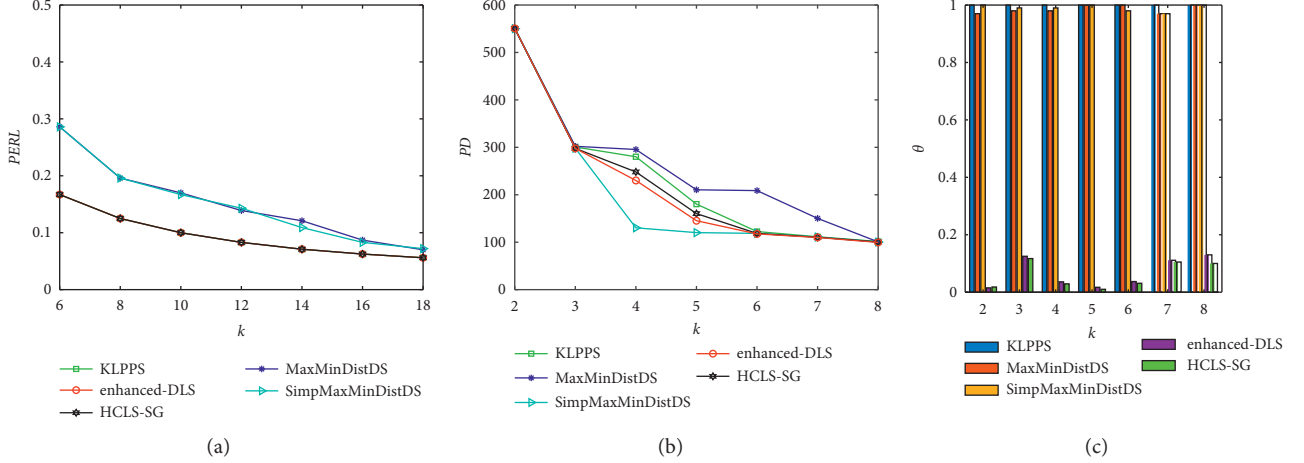


FIGURE 3: KLPPS scheme against single-point attack. (a) PERL vs. k . (b) PD vs. k . (c) θ vs. k .

so we evaluate the effectiveness of the KLPPS scheme against inference attacks from the two assessment metrics of $Q_{\text{loss}}^{\text{max}}$ and k .

(a) PL vs $Q_{\text{loss}}^{\text{max}}$

We compare the PL of KLPPS, HCLS, SG, and HCLS – SG schemes under different $Q_{\text{loss}}^{\text{max}}$ in Figure 4(a). As shown in the figure, we can draw the following 3 conclusions. First, the PL of KLPPS, SG, and HCLS – SG is significantly better than HCLS, which results from that HCLS does not consider the inference attack strategy of the adversary. Secondly, the KLPPS and HCLS – SG behave better than SG, the reason is that the former two use dummy location anonymity, increasing the difficulty of the adversary's inference. Finally, with the increase of $Q_{\text{loss}}^{\text{max}}$, the PL of the four schemes all increase, however the growth trend of PL of the four schemes slows down when reaching a certain level, which is related to the query probability distribution of user's location, indicating that the influence of $Q_{\text{loss}}^{\text{max}}$ on PL is limited.

(b) PL vs k

We compare the PL of KLPPS, HCLS, SG, and HCLS – SG schemes under different k in Figure 4(b). As we can see, three conclusions have been drawn below. First of all, as the value of k increases, the PL of KLPPS, HCLS, and HCLS – SG are significantly improved while not the SG. The reason is that SG only provides offset locations and does not consider dummy location anonymity; Secondly, the KLPPS and HCLS – SG behave better in improving PL than that of HCLS, which results from that the former two all consider the adversary's inference attack strategy, while the latter does not, so HCLS is less effective in resisting inference attacks than that of the KLPPS and HCLS – SG; Finally, the growth trend of of the four schemes slows down when reaching a certain level, which results from that the four schemes all use the offset location instead of the real location to

protect users' privacy. More specifically, the choice of l_d will make the service quality loss Q_{loss} is gradually approaching $Q_{\text{loss}}^{\text{max}}$ with the increase of k , and the four schemes are all maximized PL under the premise of ensuring that $Q_{\text{loss}} \leq Q_{\text{loss}}^{\text{max}}$, which means that when Q_{loss} gradually approaches $Q_{\text{loss}}^{\text{max}}$, the growth trend of PL slows down until $Q_{\text{loss}} = Q_{\text{loss}}^{\text{max}}$, reaching the maximum value.

(2) Q_{loss} . Q_{loss} is closely related to PL. Specifically, in some cases, users allow losing a certain service quality in exchange for higher privacy. In the experiment, we set the maximum service quality loss that the user can accept as $Q_{\text{loss}}^{\text{max}} = \{0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5\}$ and analyze the relationship between Q_{loss} and PL on this condition. Figure 5 shows the experimental results. First of all, it can be seen from the figure that the Q_{loss} and PL of the three schemes all increase with the increase of $Q_{\text{loss}}^{\text{max}}$. The reason for this is obvious: in the first place, we can see the PL will increase to a certain extent with the increase of $Q_{\text{loss}}^{\text{max}}$ from Figure 4(a). In the second place, from the definition of $Q_{\text{loss}}^{\text{max}}$ and Q_{loss} , it is obvious that Q_{loss} will increase to a certain extent with the increase of $Q_{\text{loss}}^{\text{max}}$. Secondly, under the same $Q_{\text{loss}}^{\text{max}}$, the PL of KLPPS and HCLS – SG is significantly higher than that of HCLS, but Q_{loss} is also higher than HCLS to a certain extent, which results from that both KLPPS and HCLS – SG will make full use of the limited maximum service quality loss to optimize the selection of dummy location set, so as to improve location privacy as much as possible while ensuring that the loss of service quality does not exceed the constraints of service quality, thereby effectively balancing service quality and location privacy. In addition, under the same $Q_{\text{loss}}^{\text{max}}$, the PL of KLPPS is slightly larger than that of HCLS – SG while Q_{loss} is slightly smaller than that of HCLS – SG, indicating that KLPPS is better than HCLS – SG in balancing service quality and location privacy.

The experimental results above show that the scheme can effectively resist inference attacks while effectively balancing service quality and location privacy compared with the SG, HCLS, and HCLS – SG schemes.

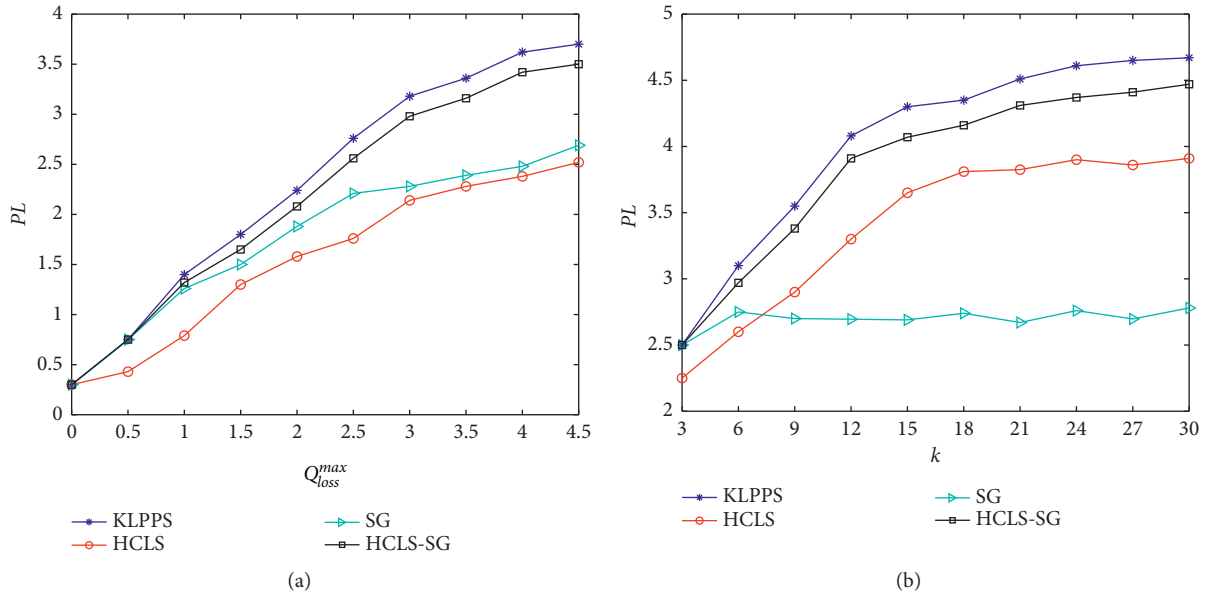


FIGURE 4: KLPPS scheme against inference attack. (a) PL vs Q_{loss}^{max} . (b) PL vs k .

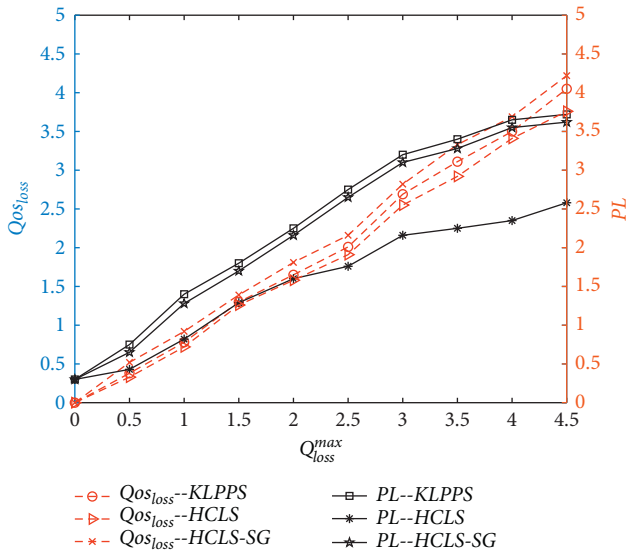


FIGURE 5: KLPPS scheme balances PL and $Q_{os_{loss}}^{max}$.

7. Conclusion

There are some problems such as single point of failure and without the ability to effectively resist single-point attack and inference attack, etc., in the traditional k -anonymous location privacy protection schemes. To solve the problems above, by analyzing the merits and drawbacks of the existing location privacy protection system architecture, we propose a semitrusted third party based location privacy protection architecture that can tackle performance bottleneck of mobile device and single point of failure. Then, comprehensively considering the characteristics of side information, semantic diversity, and physical dispersion of locations combined with the ideas of dummy location technology and offset location, a dummy location selection

algorithm based on location semantics and physical distance is proposed to effectively resist single-point attacks. Finally, we propose a location anonymous optimization method based on Stackelberg game to optimize the dummy selection algorithm. Specifically, we formalize the mutual optimization of user-adversary objectives (location privacy vs. correctness of inferring location) by using the framework of Stackelberg games, to find an optimal dummy location set. The optimal dummy location set can resist single-point attacks and inference attacks while effectively balancing service quality and location privacy. The experimental results further verify the effectiveness of the proposed scheme. However, our work still has the following shortcomings. First, in LBS, more and more people use continuous query services such as navigation services, etc., while our scheme can be only applied in snapshot query scenario not the continuous query scenario. Secondly, in different application scenarios, users have different requirements for privacy protection level and service quality, so it needs to be improved as much as possible in terms of balancing data validity and privacy levels. The next work hopes to improve our scheme to make it suitable for continuous query scenarios. Meanwhile, aiming at users with different needs in different scenarios, on the basis of further balancing service quality and privacy protection level, we design a personalized location privacy protection scheme.

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 study was supported by the Foundation of National Natural Science Foundation of China (grant number: 61962009); Major Scientific and Technological Special Project of Guizhou Province (grant number 20183001); Science and Technology Support Plan of Guizhou Province (grant number [2020]2Y011); and Foundation of Guangxi Key Laboratory of Cryptography and Information Security (grant number GCIS202118).





References

- [1] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "Psspr: a source location privacy protection scheme based on sector phantom routing in wsns," *International Journal of Intelligent Systems*, 2021.
- [2] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar, "Preserving user location privacy in mobile data management infrastructures," in *Proceedings of the International Workshop on Privacy Enhancing Technologies*, pp. 393–412, Springer, Cambridge, UK, June 2006.
- [3] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*, pp. 31–42, New York, NY, USA, May 2003.
- [4] S. Wang, Q. Hu, Y. Sun, and J. Huang, "Privacy preservation in location-based services," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 134–140, 2018.
- [5] C. S. Wright, "Bitcoin: a peer-to-peer electronic cash system," *SSRN Electronic Journal*, 2008.
- [6] T. Li, Y. Chen, Y. Wang et al., "Rational protocols and attacks in blockchain system," *Security and Communication Networks*, vol. 2020, Article ID 8839047, 11 pages, 2020.
- [7] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
- [8] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, 2021.
- [9] G. Yang, Y. Wang, Z. Wang, Y. Tian, X. Yu, and S. Li, "Ipbms: an optimal bribery selfish mining in the presence of intelligent and pure attackers," *International Journal of Intelligent Systems*, vol. 35, no. 11, pp. 1735–1748, 2020.
- [10] F. Li, Z. Liu, T. Li, H. Ju, H. Wang, and H. Zhou, "Privacy-aware PKI model with strong forward security," *International Journal of Intelligent Systems*, 2020.
- [11] Y. Chen, S. Dong, T. Li, Y. Wang, and H. Zhou, "Dynamic multi-key FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, 2021.
- [12] L. Hai, L. XingHua, L. Bin et al., "Distributed k-anonymity location privacy protection scheme based on blockchain," *Chinese Journal of Computers*, vol. 42, no. 5, pp. 942–960, 2019.
- [13] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy," in *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 617–627, Association for Computing Machinery, New York, NY, USA, October 2012.
- [14] J. Chen, K. He, Q. Yuan, M. Chen, R. Du, and Y. Xiang, "Blind filtering at third parties: an efficient privacy-preserving framework for location-based services," *IEEE Transactions on Mobile Computing*, vol. 17, no. 11, pp. 2524–2535, 2018.
- [15] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, "Location privacy-preserving mechanisms in location-based services," *ACM Computing Surveys*, vol. 54, no. 1, pp. 1–36, 2021.
- [16] S. Zhang, G. Wang, M. Z. A. Bhuiyan, and Q. Liu, "A dual privacy preserving scheme in continuous location-based services," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4191–4200, 2018.
- [17] L. Yu, L. Liu, and C. Pu, "Dynamic differential location privacy with personalized error bounds," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, USA, January 2017.
- [18] R. Shokri, "Privacy games: optimal user-centric data obfuscation," *Proceedings on Privacy Enhancing Technologies*, vol. 1, no. 2, pp. 299–315, 2015.
- [19] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of the ICPS'05. Proceedings. International Conference on Pervasive Services, 2005*, pp. 88–97, IEEE, Santorini, Greece, July 2005.
- [20] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of location privacy using dummies for location-based services," in *Proceedings of the 21st International conference on data engineering workshops (ICDEW'05)*, p.1248, IEEE, Tokyo, Japan, April 2005.
- [21] Z. Dapeng, S. Guangxuan, J. Yuanyuan, and W. Xiaoling, "Query probability-based location privacy protection approach," *Journal of Computer Applications*, vol. 37, no. 2, pp. 347–351, 2017.
- [22] L. Chang, Z. Xing, Y. Fei, L. Wanjie, and L. Shuai, "Fake location generation scheme based on user preference selection," *COMPUTER ENGINEERING AND DESIGN*, vol. 40, no. 4, pp. 914–919, 2019.
- [23] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 754–762, IEEE, Toronto, ON, Canada, May 2014.
- [24] C. Hui and Q. Xiaolin, "Location-semantic-based location privacy protection for road network," *Journal on Communications*, vol. 37, no. 8, pp. 67–76, 2016.
- [25] Z. Haiyan, Z. Kaizhong, W. Yonglu, and L. Rui, "Semantic diversity location privacy protection method in road network environment," *Computer Engineering and Applications*, vol. 56, no. 7, pp. 102–108, 2020.
- [26] Z. Yongbing, Z. Qiuyu, L. Zongyi, D. Hongxiang, and Z. Moyi, "A k-anonymous location privacy protection method of dummy based on approximate matching," *Control and Decision*, vol. 35, no. 1, pp. 55–64, 2020.
- [27] S. Chen and H. Shen, "Semantic-Aware dummy selection for location privacy preservation," in *Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 752–759, IEEE, Tianjin, China, August 2016.
- [28] W. Lu and M. XiaoFeng, "Location privacy preservation in big data era: a survey," *Journal of Software*, vol. 25, no. 4, pp. 693–712, 2014.
- [29] Q. A. Arain, I. Memon, Z. Deng, M. H. Memon, F. A. Mangi, and A. Zubedi, "Location monitoring approach: multiple mix-zones with location privacy protection based on traffic flow over road networks," *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 5563–5607, 2018.

- [30] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, IEEE, Atlanta, GA, USA, May 2017.
- [31] X. XingYou, B. ZhiHong, L. Jie, and Y. RuiYun, "A location cloaking algorithm based on dummy and stackelberg game," *Chinese Journal of Computers*, vol. 42, no. 10, pp. 2216–2232, 2019.
- [32] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 251–262, Association for Computing Machinery, New York, NY, USA, November 2014.
- [33] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Proceedings of the 2011 IEEE symposium on security and privacy*, pp. 247–262, IEEE, Oakland, CA, USA, May 2011.
- [34] W. Zhen, Y. Yong, A. Bo, L. MingChu, and W. FeiYue, "An overview of security games," *Journal of Command and Control*, vol. 1, no. 2, pp. 121–149, 2015.
- [35] W. Sheng, L. Fenghua, N. Ben, S. Zhe, and L. Hui, "Research progress on location privacy-preserving techniques," *Journal on Communications*, vol. 37, no. 12, pp. 124–141, 2016.

Research Article

A Regulatable Data Privacy Protection Scheme for Energy Transactions Based on Consortium Blockchain

Yufeng Li ¹, Yuling Chen ¹, Tao Li ¹ and Xiaojun Ren ²

¹State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China

²Blockchain Laboratory of Agricultural Vegetables, Weifang University of Science and Technology, Shouguang 262700, China

Correspondence should be addressed to Tao Li; litao_2019@qfnu.edu.cn

Received 6 October 2021; Accepted 10 November 2021; Published 7 December 2021

Academic Editor: Chien Ming Chen

Copyright © 2021 Yufeng Li 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 blockchain-based energy transaction scenario, the decentralization and transparency of the ledger will cause the users' transaction details to be disclosed to all participants. Attackers can use data mining algorithms to obtain and analyze users' private data, which will lead to the disclosure of transaction information. Simultaneously, it is also necessary for regulatory authorities to implement effective supervision of private data. Therefore, we propose a supervisable energy transaction data privacy protection scheme, which aims to trade off the supervision of energy transaction data by the supervisory authority and the privacy protection of transaction data. First, the concealment of the transaction amount is realized by Pedersen commitment and Bulletproof range proof. Next, the combination of ElGamal encryption and zero-knowledge proof technology ensures the authenticity of audit tickets, which allows regulators to achieve reliable supervision of the transaction privacy data without opening the commitment. Finally, the multibase decomposition method is used to improve the decryption efficiency of the supervisor. Experiments and security analysis show that the scheme can well satisfy transaction privacy and auditability.

1. Introduction

With the birth of Bitcoin [1], blockchain, as the underlying technology of Bitcoin [2], has been widely used in finance, medical data sharing, supply chain, energy trading, and other fields. Blockchain has the advantages of decentralization, tamper-proof, autonomy, traceability, and so on, which is regarded as the future of infrastructure. The subsequent emergence of Ethereum means that smart contracts can be used to settle the problem of decentralized applications in the currency fields [3, 4]. At present, the application mode of blockchain can be divided into three categories, public chain, consortium blockchain, and private chain. The public chain allows users to enter and exit freely, while the consortium blockchain and private chain require authorization and verification before joining. The consortium blockchain is a blockchain composed of multiple institutions. The designated members of the consortium blockchain participate in the consensus process and the

maintenance of the ledger. The consortium blockchain has the advantages of fast transaction processing speed and high transaction efficiency. Therefore, it is widely used in energy trading, commodity traceability, supply chain management, and other fields. In the field of energy trading, blockchain technology is used to integrate scattered energy nodes to establish a distributed energy trading platform based on P2P transactions. It does not require third-party intermediaries and provides a low-cost trading platform for transactions between distributed energy nodes. The most important feature is that it can reasonably settle the trust problem in distributed energy transactions. Simultaneously, the blockchain-based transaction model can promote the fairness and openness of transactions in the Energy Internet and accelerate the circulation of data elements.

There are still some shortcomings existing in the practical application of energy trading. Specifically, miners and verification nodes in the blockchain can quickly verify the legitimacy of transactions due to the openness and

transparency of the ledger. However, information such as the users' identity and transaction details will be disclosed to all participants of the network in the process. Moreover, the external attackers [5, 6] can obtain information such as the account, geographic location, energy usage, and source location of the energy node from the transaction record [7]. When obtaining such information, attackers can predict users' next behavior by data mining, data analysis, machine learning, and other methods [8, 9]. Therefore, in the scenario of distributed energy transactions based on blockchain, the issue of data privacy protection in energy transactions has gradually become a new challenge. In transactions, privacy protection issues are mainly divided into two categories: identity privacy and transaction data privacy issues. Identity privacy means that attackers cannot obtain any useful information related to their identity only through the content of public data stored on the chain. Transaction data privacy refers to the fact that both parties to the transaction are considering their interests, and any node other than themselves cannot obtain the details of the transaction from public information. The contributions of this paper are as follows:

- (1) There are two problems in the blockchain-based energy transaction scenario. The openness and transparency of the transaction ledger allow any participant to obtain transaction details, which poses the risk of private data leakage. Simultaneously, there needs to be a balance between transaction regulation and privacy protection. Therefore, we propose a supervisable energy transaction data privacy protection scheme to deal with the above problems.
- (2) Combine Pedersen commitment and Bulletproof range proof to realize the concealment of the transaction amount. Adopt ElGamal encryption and zero-knowledge proof technology to ensure the authenticity of audit tickets. The reliability of the transaction can be supervised without executing the open commitment. The introduction of multibase decomposition technology in ElGamal improves the decryption efficiency of the supervisor.
- (3) Security and performance analysis show that the scheme can audit a certain transaction or multiple transactions in the ledger and effectively protect the privacy of transaction amounts.

2. Related Works

At present, blockchain technology is developing rapidly, and the privacy protection issue in the blockchain has received extensive attention from a growing number of scholars. A variety of cryptographic technologies are applied in the blockchain system to settle the problems of identity privacy and transaction privacy, which also means that the supervision technology of blockchain transactions will face more challenges.

The cryptocurrency based on the public chain emphasizes the privacy protection of transactions. For example, these works [10, 11] proposed a Mixcoin protocol, which uses a Mixcoin protocol to transfer funds from multiple input

addresses to multiple output addresses to provide anonymity services. The connection between the user's real identity and address was interrupted. In Monero [12], the Pedersen commitment scheme is used to conceal transaction information. It uses ring signatures and one-time addresses to hide the identities of the sender and receiver in the transaction. Based on Monero, Li et al. [13] proposed a new cryptocurrency system, which can simultaneously achieve identity anonymity and traceability in Monero. However, excessive privacy protection strategies will cause the regulatory authorities to not effectively supervise the transaction content and identity. Zcash [14] uses noninteractive zero-knowledge proofs (zk-SNARKs) technology to verify private transactions and conceal the identity of the sender. However, the transaction efficiency of this scheme is unsatisfactory. An anonymous scholar named Tom Elvis Jedusor first proposed the MimbleWimble protocol in 2016 [15]. It uses confidential transaction technology to realize the shielding of transaction content and realizes the concealment of the identity of the transaction party by removing the transaction address. Although the agreement has regulatory functions, it cannot track transaction information and the identity of violators. In 2019, Beam and Grin were proposed. The scheme combines the MimbleWimble protocol and aggregated signature technology to achieve the purpose of protecting the privacy of blockchain transactions [16]. These works [17, 18] proposed a blockchain-based machine learning framework and secure key management scheme (BC-EKM). This scheme designs a secure cluster formation algorithm and a secure node movement algorithm to implement key management, where stake blockchain as a trust machine replaces the majority functions of the BS. In addition, this scheme is based on the SM2 public-key cryptosystem to protect data security and prevent data privacy leakage in edge services. Chen et al. [19] proposed a new ciphertext extension method that makes homomorphic encryption of ciphertext more efficient. However, the scheme requires both parties to the transaction to interact online, which will encounter difficulties in practice.

The above privacy protection scheme is a typical public chain application scenario. The privacy protection features they provide do not implement transaction supervision and cannot satisfy the supervision requirements of the application system. Therefore, privacy protection schemes with supervisory functions have also been proposed. Wüst et al. [20] proposed a new cryptocurrency PRCash in 2018. It uses zero-knowledge proof technology to generate range proof and regulatory proof for each transaction. The range proof is verified by the public node, which is used to guarantee the range of the user's transaction amount. The supervision certificate is verified by the supervisor, which is used to restore user identity information. The regulator in PRCash supervises the total amount of transactions made by users over a while. If the user's total transaction amount exceeds the quota specified by the system within a certain period, the supervisor can track the violating user. The supervisor obtains its true identity information based on the supervisory certificate. PRCash realizes the limitation of the user's transaction amount within a period and supervises violations, but it cannot obtain the specific value of each transaction. NeHa et al. [21] proposed a

comprehensive privacy auditable distributed ledger system Zkledger in 2018. The program uses a table ledger structure, which can conceal the identities of the sender and receiver of the transaction and the transaction amount simultaneously. Zkledger has set up a supervisor, and the supervisor needs to initiate an online inquiry to the user to obtain the sum of the user's assets over a while. However, the regulator cannot obtain the specific amount of each transaction during the entire process. Moreover, the user needs to open the commitment after responding to the supervision request, and the supervisor will obtain certain commitment secret value information in the process, which is not conducive to the security of the system. In 2019, Kang et al. [22] proposed the privacy protection smart contract Fabzk based on Zkledger. This scheme assigns the five zero-knowledge proofs generated in the transaction to system users and supervisors for verification. To improve transaction performance, the transaction verification process can be performed concurrently. However, the scheme requires the regulator to remain online at all times, and the transaction is considered valid only if all five verification equations are passed.

Regarding the transaction privacy issues in the energy transaction scenario based on the consortium blockchain, this paper proposes a supervisable energy transaction data privacy protection scheme. This scheme realizes the concealment of the transaction amount by the Pedersen commitment and uses the Bulletproofs range proof to guarantee the transaction amount range. Combining ElGamal encryption and zero-knowledge proof technology to ensure the authenticity of regulatory tickets, the regulation of transactions can be achieved without executing open commitments. The multibase decomposition technology is introduced in the scheme to improve the decryption efficiency of the regulator. The results of experiments and security analysis show that the scheme can achieve transaction privacy and auditability. The supervisor can audit the total transaction amount in a certain number of blocks, and it can also restore the specific amount in a transaction.

3. Preliminaries

3.1. Consortium Blockchain. Blockchain is a new technology system derived from the underlying technology of Bitcoin. Blockchain technology is developing rapidly, and it has derived consortium blockchain and private chains from public chains. The public chain is completely decentralized, and any user can join or exit freely, while the private chain is a completely private blockchain, and only internal personnel can use it. The degree of decentralization of the consortium blockchain is between the public chain and the private chain. It is mostly composed of offline enterprises and other alliances. Users need to achieve certain conditions and obtain permission to enter and exit. Additionally, the consortium blockchain can be completely open or only accessible by insiders of the consortium.

3.2. Zero-Knowledge Proof. The zero-knowledge proof system involves two parties, called the prover and verifier. Prover knows a certain secret, and prover hopes to convince

verifier that he does have the secret without revealing the secret. The zero-knowledge proof system should satisfy the following three conditions: completeness, reliability, and zero-knowledge. Completeness means that if the prover knows a certain secret, the verifier will accept the prover's proof. Reliability means that if the prover can convince the verifier with a certain probability, the prover knows the corresponding secret. Zero-knowledge refers to the fact that the verifier cannot obtain any additional information during the interaction between the prover and the verifier. Zero-knowledge proofs can be classified into interactive and noninteractive knowledge proofs. Interactive zero-knowledge proof requires one or more communications between the prover and the verifier. Blum et al. [23] proposed a noninteractive zero-knowledge proof. The prover uses hash value instead of the interactive process, which avoids multiple communications between the prover and the verifier. Typical representatives of noninteractive zero-knowledge proof protocols are Bulletproofs [24] and ZK-SNARKs [25]. Bulletproofs have the characteristics of short proof time and no need to set up a trusted center.

Comparing the noninteractive zero-knowledge proof with the interactive zero-knowledge proof, the noninteractive zero-knowledge proof avoids multiple rounds of communication between participants. And any participant can verify the validity of the proof π . The Fiat-Shamir [26] scheme provides a method to transform interactive zero-knowledge proofs into noninteractive zero-knowledge proofs. This feature fits perfectly with the decentralized environment of the blockchain, which can reach a consensus and establish a trust relationship between nodes that do not trust each other.

3.3. Pedersen Commitment. The cryptographic commitment scheme is a two-stage interactive protocol involving two parties, and the two parties are the promiser and the receiver, respectively. The first stage is the commitment stage. The promiser chooses a message m and sends it to the receiver in the form of ciphertext, which means that it will not change m . The second stage is the opening stage, where the promiser discloses the message m and the blinding factor, and the receiver uses this to verify whether it is consistent with the message received in the promise stage. The commitment scheme has two basic properties: hiding and binding. Hiding is the commitment and will not reveal any information about the message m . Binding means that no malicious promiser can open the commitment to m and pass the verification, which means that the receiver can be sure that m is the message corresponding to the commitment. Pedersen promises to be an important cryptographic component in blockchain technology, and its structure consists of the following three stages:

Setup: select the elliptic curve $E(F_p)$ with G and H , where G and H are the two generators of the elliptic curve, and the order is q . Public parameters are (G, H, q) .

Commitment: the promiser chooses a random number k as the blind factor, calculates the commitment

Com = $kG + vH$, and then sends the commitment Com to the receiver.

Open: the promiser sends (v, k) to the receiver, and the receiver verifies whether the commitment is equal to $kG + vH$ and accepts if they are equal; otherwise, it rejects the commitment.

The homomorphic characteristics of Pedersen commitment are embodied as follows: $\text{Com}(v_1) + \text{Com}(v_2) = (k_1 + k_2)G + (v_1 + v_2)H = \text{Com}(v_1 + v_2)$. According to this feature, the verifier can calculate the transaction commitment without knowing the specific secret.

3.4. Elliptic Curve Cryptography. Elliptic curve cryptography was first proposed by Neal Koblitz and Victor Miller in 1985, and it is called ECC for short. It is a public-key cryptosystem that is currently widely used. The security of the ECC algorithm is mainly based on the Elliptic Curve Discrete Logarithm Problem (ECDLP). Under the same security requirements, its required parameters and key size are shorter. Compared with other public-key cryptosystems, elliptic curve cryptography has the advantages of higher security, short key length, small storage space, and fast calculation speed.

Let Z_p denote the domain of integers, where p is a large prime number; thus, an elliptic curve $E(F_p)$ can be defined. It can usually be expressed as $y^2 = x^3 + ax + b \pmod{p}$, where the coefficients $a, b \in Z_p$. a and b are two constants satisfying $4a^3 + 27b^2 \pmod{p} \neq 0$. $P = (x, y)$ represents a point on the elliptic curve, where $x, y \in Z_p$ represents the abscissa and ordinate of the corresponding point on the elliptic curve, respectively. There is a special point O on the elliptic curve, called the point of infinity, which forms the elliptic curve $E(F_p)$ together with the whole point P .

3.5. ElGamal Encryption Algorithm. ElGamal encryption is a common asymmetric encryption algorithm. Its security is based on the finite field discrete logarithm problem [27], and it is indistinguishable under selected plaintext attacks (IND-CPA). ElGamal encryption mainly includes three algorithms: key generation algorithm, encryption algorithm, and decryption algorithm.

Key generation algorithm: select the finite field cyclic group G of order p , where p is a large prime number. The generator of the finite field cyclic group G is g . Randomly select $x \in Z_p$ as the private key, calculate the public key $y = g^x \pmod{p}$, and make it public.

Encryption algorithm: the encrypting party chooses a plaintext message m , and the plaintext message m needs to satisfy $m < p$ and then choose a random number $k < p$, where k and $p - 1$ are relatively prime. Calculate the ciphertext $A = g^k \pmod{p}$ and $B = my^k \pmod{p}$. The ciphertext consists of two parts $C = (A, B)$.

Decryption algorithm: the decryptor uses his private key x to decrypt the ciphertext (A, B) and restore the plaintext by calculating $m = B/A^x \pmod{p}$.

In addition, the relevant symbols and explanations involved in this paper are listed in Table 1.

4. Supervisable Privacy Protection Scheme Model

The scheme satisfies the primary principles of privacy and supervisability of transactions, which are required to prevent the leakage of sensitive user information to ensure transaction privacy. The Pedersen commitment based on the elliptic curve is applied to the scheme to hide the transaction information, and the zero-knowledge range proof ensures that the transaction amount hidden in the commitment is in the legitimate interval. We combine homomorphic encryption technology, Pedersen commitment, and zero-knowledge proof technology to ensure that the transaction amount is consistent with the amount in the ciphertext. To further improve transaction performance, multibase decomposition technology is used to improve the efficiency of encryption and decryption by regulatory authorities. Based on these cryptography technologies, we have designed a regulatory privacy protection scheme to achieve privacy and regulations.

4.1. Transaction Structure. There are mainly five entities in this program, as shown in Figure 1, which are the certification body (CA), regulatory (RA), energy aggregator, energy buyer EB, and energy seller ES. The role of each entity is as follows:

Certification authority (CA): Its role is to issue a certificate for the user. Any user who wants to enter the blockchain network must be authorized by the certification authority and obtain the certificate *Cert* promulgated by the certification authority to the certificate *Cert*.

Regulatory authority (RA): It is responsible for auditing transaction content. Once suspicious transactions are found, the supervisor can obtain ciphertext and decrypt the specific one after the transaction information is decrypted from the transaction information and interact with the CA to obtain real-name information of the transaction. It is worth noting that this scheme can be audited in this scheme, which means that there is no need to travel online.

Energy buyers (EB): It uses the initiator of the transaction to launch a transaction as a sender in the transaction. It uses the private key to sign the transaction proposal during the transaction process, which is a transaction commitment, supervision ciphertext, and zero-knowledge used to prove the effective effectiveness. Before generating a complete transaction, it is necessary to make a chain interaction with the recipient of the transaction, and the purpose is to consent with the two parties. Moreover, in the actual transaction process, ES can also initiate a transaction as a transaction sender. To describe convenience, only EB is considered as a sender, and ES is considered as a reception party.

TABLE 1: Nomenclature.

$E_p(a, b)$	An elliptic curve with parameters a, b, p
H, G, n	Generator and order
P_a, x_a	Supervisor public key and private key
params	Public parameters
v	Transaction amount
k, r	Blinding factor and random number
$Rf(v)$	Range proof of transaction amount
$C(k, v)$	Transaction commitment
bill	System audit ticket
S	Transaction balance signatures
e, σ	Challenge value
Cert	Identity certificate
C_x, D_x, B_x	Regulatory ciphertext
π	Zero-knowledge proof
V_x, E_x, T_x	ZK auxiliary information
η_x, k_x, r_x	An element in vectors η, k, r

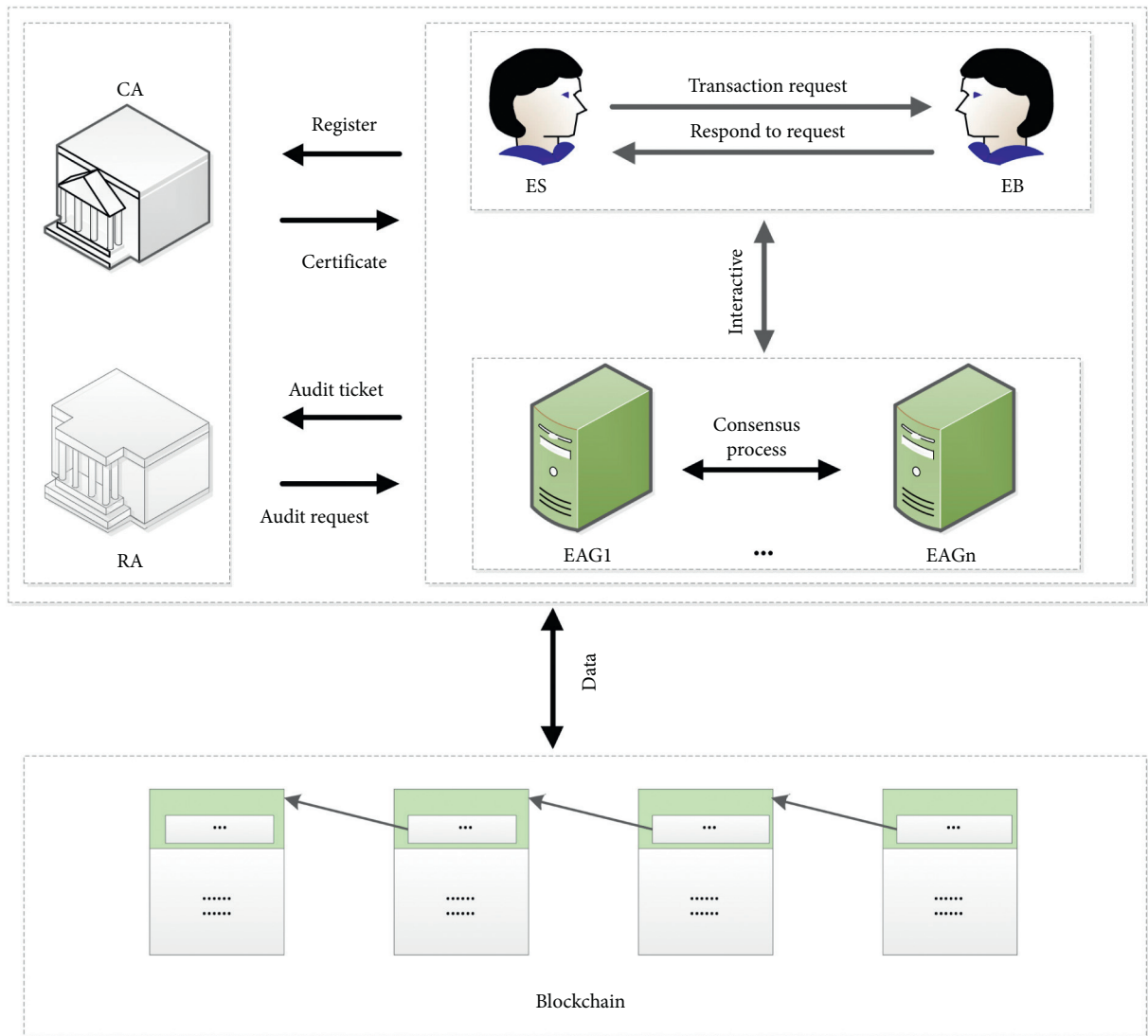


FIGURE 1: Scheme architecture.

Energy seller (ES): It serves as a receiver of the transaction. During the generation of the transaction, there is an interactive process between the recipient and the sender to generate some interaction information.

EAG: It sends an audit request to the user, verifies the user's audit notes, and interacts with the supervisor.

4.2. Transaction Process. The transaction process consists of two stages: system initialization and interaction between buyers and sellers.

4.2.1. System Initialization. Our scheme is combined with the Pedersen commitment and ElGamal encryption technology. The system parameters are needed to generate a scheme in the initialization phase. Select an elliptic curve $E_p(a, b)$, one generating element G on the curve, and its order is n . Randomly select k , $x_a < n$, calculate $H = kG$ and $P_a = x_a G$, and the system deletes the discrete logarithm k of H . Therefore, the discrete logarithm of H is unknown to the outside world. We use x_a as the private key of the regulator and P_a as the public key of the regulator. Finally, the public parameters may be represented as $\text{params} = (E_p(a, b), n, G, H, P_a)$.

4.2.2. Interactive Process. In the energy transaction payment phase, EAG returns the results of the bid to energy buyer A and energy seller B. v_3 is the transaction price between buyer A and seller B. Subsequently A checks if the amount in all the addresses is greater than the transaction amount v_3 ; otherwise, the trading will be terminated. Assume that the amounts v_1 and v_2 in the two addresses of A are satisfied with $v_3 = v_1 + v_2$, where $v_1, v_2, v_3 \in [0, 2^n - 1]$. The detailed trading steps are as follows:

Step 1: A initiates a transaction to B. Specifically, A pays v_1 and v_2 from B, which is the amount that matches in advance. A selects blind factors $k_1, k_2 < n$ and a random number $r_a < n$ at random and calculates the necessary information and range proof. Among them, the range proof proves that $Rf_{in1}(v_1)$ and $Rf_{in2}(v_2)$ are generated by Bulletproof technology. The specific calculation process is as follows:

$$R_a = r_a G, \quad (1)$$

$$K_a = (k_1 + k_2)G, \quad (2)$$

$$C_{in1}(k_1, v_1) = v_1 H + k_1 G, \quad (3)$$

$$C_{in2}(k_2, v_2) = v_2 H + k_2 G. \quad (4)$$

Then compose the above result into m_1 . And send $m_1 = (\text{params}, K_a, R_a, Rf_{in1}, Rf_{in2}, C_{in1}, C_{in2})$ and certificate $Cert_A$ to B by the secure channel.

Step 2: After receiving the transaction request, B immediately verifies the legitimacy of the certificate, whether the scope certification and commitment are

correct. If any verification fails, the transaction is terminated. Among them, the verification commitment is equivalent to the verification equation (5). If the verification is passed, it means that the transaction initiator has correctly calculated the commitment according to the rules. B randomly selects the blinding factor and random number $k_3, r_b < n$ and calculates the commitment $C_{out}(k_3, v_3)$, the range proof $Rf_{out}(v_3)$, the transaction balance signature S_b , and the audit ticket bill_{out} (the construction of the audit ticket will be detailed in Section 4.4). The specific calculation process is as follows:

$$C_{in1}(k_1, v_1) + C_{in2}(k_2, v_2) = v_3 H + K_a, \quad (5)$$

$$R_b = r_b G, \quad (6)$$

$$K_b = k_3 G, \quad (7)$$

$$C_{out}(k_3, v_3) = v_3 H + k_3 G, \quad (8)$$

$$\begin{aligned} K &= K_a + K_b, \\ R &= R_a + R_b, \end{aligned} \quad (9)$$

$$e = \text{Hash}(\text{params}, R, K), \quad (10)$$

$$S_b = r_b + e k_3, \quad (11)$$

$$\text{bill}_{out} = (C_{out}, D_{out}, \pi_{out}). \quad (12)$$

Then compose the above result into m_2 . And send $m_2 = (\text{params}, K_b, R_b, Rf_{out}, C_{out}, S_b, e, \text{bill}_{out})$ and certificate $Cert_B$ to A by the secure channel.

Step 3: when A accepts and receives the reply, it will verify the validity of the certificate. If the verification is passed, then extract K_b and R_b from the message m_2 to calculate $K = K_a + K_b, R = R_a + R_b$. Verify whether equation (13) is established. If the verification is passed, calculate the transaction balance signature S and audit bill $\text{bill}_{in1}, \text{bill}_{in2}$; otherwise, terminate the transaction. The specific calculation process is as follows:

$$\text{Hash}(\text{params}, R, K) = e, \quad (13)$$

$$S_a = r_a + e(k_1 + k_2), \quad (14)$$

$$S = S_a + S_b, \quad (15)$$

$$\text{bill}_{in1} = (C_{in1}, D_{in1}, \pi_{in1}), \quad (16)$$

$$\text{bill}_{in2} = (C_{in2}, D_{in2}, \pi_{in2}). \quad (17)$$

Finally, combine the above results into a private transaction $Tx = (\text{params}, (C_{in,i}(k_i, v_i), Rf_{in,i}(v_i), \text{bill}_{in,i})_{i \in [1,2]}, (C_{out,j}(k_j, v_j), Rf_{out,j}(v_j), \text{bill}_{out,j})_{j=1}, R, K, S, e, Cert_A, Cert_B)$ and send Tx to EAG.

Step 4: EAG will verify its correctness after receiving the Tx . It mainly includes the legality of the certificates $Cert_A$ and $Cert_B$, the correctness of the scope certification, and the correctness of the signature (S, e) . Among them, verifying the correctness of the signature is equivalent to verifying whether the equation $SG = R + eK$ is established, which means that the sum of the input of the transaction is equal to the sum of the output; otherwise, the transaction is discarded.

4.3. Supervision Process. The supervision process mainly consists of three entities: RA, EAG, and CA. (1) RA: it has supervision and audit functions. In the supervision process, the specific amount in a certain transaction can be audited. (2) EAG: it provides audit-related transaction information for regulators. (3) CA: when the supervisor finds that the transaction is abnormal, it can extract the certificate $Cert$ from the transaction information and interact with the CA to trace the identity of the trader. Specifically, it consists of the following steps:

Step 1: Verify the correctness of the zero-knowledge proof in the audit ticket. After submitting an audit request to EAG, the supervisor obtains a transaction Tx and extracts the audit bill $bill_{in,i}, bill_{out,j}$ from it. For the convenience of description, we simplified the audit ticket as $bill = (C, D, \pi)$. For the zero-knowledge proof $\pi = PK\{(\eta_x, k_x, r_x)_{x \in [0, l-1]}: C_x = \eta_x H + k_x G \wedge D_x = \eta_x H + r_x P_a \wedge B_x = r_x G\} = \{(Z_{\eta_x}, Z_{k_x}, Z_{r_x})_{x \in [0, l-1]}, \sigma\}$ (its generation process will be described in detail in Section 4.4), calculate the following values, respectively:

$$E_x = Z_{r_x} G - \sigma B_x, \quad (18)$$

$$T_x = Z_{\eta_x} H + Z_{r_x} P_a - \sigma D_x, \quad (19)$$

$$V_x = Z_{\eta_x} H + Z_{k_x} G - \sigma C_x, \quad (20)$$

$$\sigma' = H(\text{params}, (V_x, E_x, T_x)_{x \in [0, l-1]}). \quad (21)$$

Verify that equation $\sigma' = \sigma$ is established. If the verification is passed, it means that the ciphertext and the commitment calculation in the audit ticket are correct. Otherwise, it means that there are violating nodes participating in the transaction, which requires interaction with the CA through the certificate in the transaction information to track the identity of the suspicious transaction initiator.

Step 2: The supervisor uses its private key x_a to decrypt the ciphertext to obtain the specific transaction amount. To improve the efficiency of encryption and decryption, the supervisor precomputes a table $(0H, 1H, \dots, (u-1)H)$ and stores it locally. The supervisor calculates $y_x = D_x - x_a B_x$ by extracting the ciphertext $D = (B_x = r_x G, D_x = \eta_x H + r_x P_a)_{x \in [0, l-1]}$ from the audit ticket. According to y , the auditor uses a precomputation table containing t to find out the value

of η_x . Finally, the specific amount in each transaction is restored by calculating $v = \sum_{x=0}^{l-1} \eta_x u^x$.

4.4. Construction of Audit Ticket. This section will describe in detail the construction of an audit ticket. For large transaction amounts, to improve supervision efficiency and system performance, multibase decomposition is used to achieve efficient decryption of ciphertext by regulatory agencies. The generation of audit tickets consists of the following three steps:

Step 1: Decompose a transaction amount v into a set of vectors $\eta = (\eta_0, \dots, \eta_{l-1})$, $\eta_x \in [0, u-1]$, where u represents the basis of multibase decomposition, satisfying $v = \sum_{x=0}^{l-1} \eta_x u^x$.

Step 2: For each element η_x in the vector η , calculate the ElGamal ciphertext $D = (B_x = r_x G, D_x = \eta_x H + r_x P_a)_{x \in [0, l-1]}$ and the commitment $C = (\eta_x H + k_x G)_{x \in [0, l-1]}$, where r_x and k_x satisfy $r = \sum_{x=0}^{l-1} r_x u^x$ and $k = \sum_{x=0}^{l-1} k_x u^x$.

Step 3: For each element η_x in the vector η , calculate the zero-knowledge proof $\pi = PK\{(\eta_x, k_x, r_x)_{x \in [0, l-1]}: C_x = \eta_x H + k_x G \wedge D_x = \eta_x H + r_x P_a \wedge B_x = r_x G\}$. The specific details are as follows: randomly select v_x, t_x, s_x , calculate $V_x = v_x H + t_x G, E_x = s_x G, T_x = v_x H + s_x P_a$, calculate $\sigma = H(\text{params}, (V_x, E_x, T_x)_{x \in [0, l-1]})$, and calculate $Z_{\eta_x} = v_x + \sigma \eta_x, Z_{k_x} = t_x + \sigma k_x, Z_{r_x} = s_x + \sigma r_x$. Obtain a zero-knowledge proof $\{(Z_{\eta_x}, Z_{k_x}, Z_{r_x})_{x \in [0, l-1]}, \sigma\}$ with a transaction output amount of v . Finally, we get the audit bill $bill = (C, D, \pi)$.

5. Security Analysis

5.1. Security Requirements. The security goals of the scheme will be defined as follows: (1) Transaction balance: it means that the total input of a transaction is equal to the total output, which means that users cannot create or destroy a transaction arbitrarily. (2) The privacy of the transaction: except for the parties to the transaction and the supervisor, other users cannot obtain specific information about the transaction amount based on public information such as transaction balance signatures, commitment values, and audit tickets. (3) Auditability of transactions: when the supervisor needs to review a certain transaction or multiple transactions in a certain block, the supervisor can audit the corresponding transaction amount and trace the user identity.

5.2. Analysis

5.2.1. Transaction Balance. Suppose H is a random oracle. If the discrete logarithm problem of transaction balance signature is difficult and the commitment scheme satisfies the binding property, then this scheme satisfies the transaction balance.

The proof process is an interactive game between the algorithm opponent A and the mathematical problem

opponent B. B receives a random DLP problem instance $H = xG$, and his goal is to calculate x . B uses A as a subroutine to calculate x , and the mathematical problem opponent B plays the challenger of the algorithm opponent A.

System initialization phase: B sends the system public parameters params to A. B has to maintain two tables L_c and L_s , which are empty at the initial moment. L_c is used to simulate the query of the algorithm opponent A on the commitment value and L_s is used to simulate the transaction balance signature query.

Inquiry stage: the algorithmic opponent A, respectively, inquires the commitment $C(v_i)$ and the transaction balance signature S_i to the commitment oracle and the transaction balance signature oracle for a limited number of times. If there is no corresponding value in L_c and L_s , B randomly selects the parameter to calculate the corresponding value, returns it to A, and updates L_c and L_s .

Forgery stage: suppose the algorithm opponent A successfully forged a transaction $(C(v'), K', R', S', e')$ by the above query and the forged transaction balance signature is (S', e', K') , where $K' = v'H + e'G + K$. It can be verified that the equation $S'G = R' + e'K'$ holds. During the interrogation process, the adversary of the algorithm also obtains a correct signature (S, e, K) and can also verify that the equation $SG = R + H(R, K)K$ is established.

The discrete logarithm x corresponding to H can be solved by combining the above two equations. It can be seen that algorithmic opponent A can successfully break the trading balance. Mathematical problem opponent B can use A to settle the discrete logarithm problem, which contradicts the DLP assumption of this scheme. Therefore, this scheme satisfies the transaction balance.

5.2.2. The Privacy of Transactions. Transaction input and output are stored in the blockchain in the form of commitment. Because the blinding factor is unknown and the discrete logarithm is difficult, other participants cannot know the specific amount of the transaction except for the two parties in the transaction. Simultaneously, EAG will mix all inputs and outputs, which breaks the logical connection between the transaction input address, transaction output address, and change address, thereby ensuring the privacy of transactions.

5.2.3. Auditability of Transactions. This scheme uses Bulletproof to ensure that the transaction amount is within a specific range. The supervisor can obtain the specific details of a transaction by its private key. When the supervisor needs to verify the information of a certain transaction, it uses the private key to decrypt the audit ticket to obtain detailed information of a certain transaction. If suspicious behavior is discovered, the user's identity information can be obtained by interaction with CA for accountability.

6. Performance Analysis

We analyze our scheme based on throughput and latency. Throughput and latency are the two most important indicators for analyzing the performance of a blockchain system.

We analyze this scheme based on these two indicators. Transaction delay and throughput are affected by transaction zero-knowledge proof generation time, verification time, audit ticket generation time, supervisor audit time, and transaction size. The comparative analysis between this scheme and the existing scheme is shown in Table 2.

6.1. Experimental Configuration. This experiment was conducted on a computer with 8G memory, Inter (R) Core (TM) i5-65003.20 GHz CPU and GeForce GT 730 graphics card, and 64-bit Windows10 operating system. The scheme is implemented in C language, in which the Hash algorithm uses the cryptographic hash algorithm SM3, and the elliptic curve selects the more efficient SM2. We set $n = 64$; that is, we use a 64-bit positive integer to represent the transaction amount. This setting is to be the same as Bitcoin and Monero. In our experiments, we mainly consider the following two aspects, time overhead and storage overhead. Storage overhead is mainly the size of a transaction. Time overhead mainly includes transaction generation time, verification time, and audit time. We compare the above indicators with another similar scheme.

6.2. Results and Analysis. The time cost and storage cost of a single transaction mainly consider the classic transaction scenario of 2 inputs and 2 outputs. We compared our scheme with similar schemes, as shown in Table 3.

In our scheme, the size of the audit ticket is approximately 320 bytes. For each output, the range proved to be about 738 bytes in size. The committed size for each transaction is 64 bytes. The transaction generation time cost is about 90 ms, the transaction verification time cost is about 10 ms, and the transaction audit time cost is about 43 ms.

For [12], it uses ring signature technology. For more convenience, we set the size of the ring signature to 4. In a transaction with 2 inputs and 2 outputs, its total storage cost is 12710 bytes, and the time cost of the sender and verifier is approximately 300 ms.

For [14], all the proofs can be combined into a 288-byte zk-SNARK proof through aggregation technology. The total proof size of a classic transaction with 2 inputs and 2 outputs becomes about 576 bytes. Since the zk-SNARK proof is adopted, the time cost of generating the proof will be very large, about 2 minutes. And zk-SNARK proves that a large amount of memory RAM $>$ 3 GB needs to be consumed during the generation process. It will cause long delays in the blockchain system. However, the verification time overhead of the zk-SNARK proof is considerable, about 10 ms.

For the privacy of transactions, we set 64-bit transaction data, which means that the data range is $v \in [0, 2^{64} - 1]$. We use the multibase decomposition method to improve the efficiency of encryption and decryption, taking $u = 2^8 = 256$, $l = 8$. Figures 2 and 3 show the comparison of encryption time and decryption time between Elgamal and MBD_Elgamal (Elgamal encryption based on multibase decomposition). From the figure, we can see that, with the increase of transaction data length, the encryption efficiency

TABLE 2: Scheme comparison.

Schemes	Main technique	Transaction privacy	Offline supervision	Public verification	Single transaction audit
[12]	CryptoNote protocol Ring signature	Yes	No	No	No
[14]	zk-SNARKs Pedersen commitment	Yes	No	No	Yes
This paper	Pedersen Commitment, Bulletproofs, and ElGamal	Yes	Yes	Yes	Yes

TABLE 3: Comparison of privacy-preserving blockchain schemes.

Schemes	Storage overhead (bytes)	Transaction time	Verification time	Audit time
[12]	12710	300 ms	300 ms	—
[14]	576	120 s	10 ms	55 ms
This paper	4488	90 ms	10 ms	43 ms

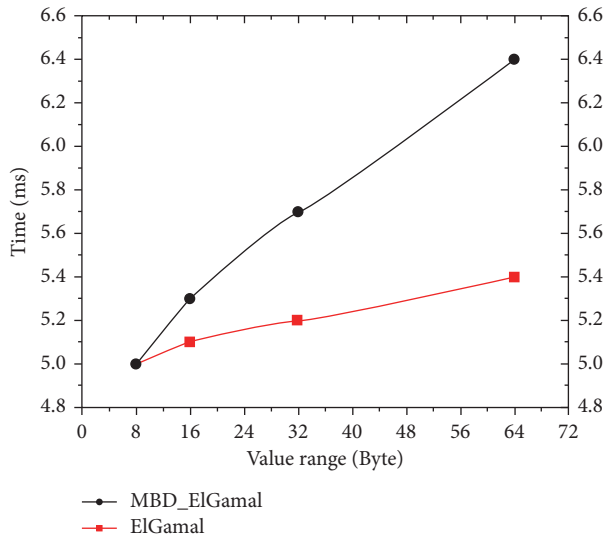


FIGURE 2: Encryption time.

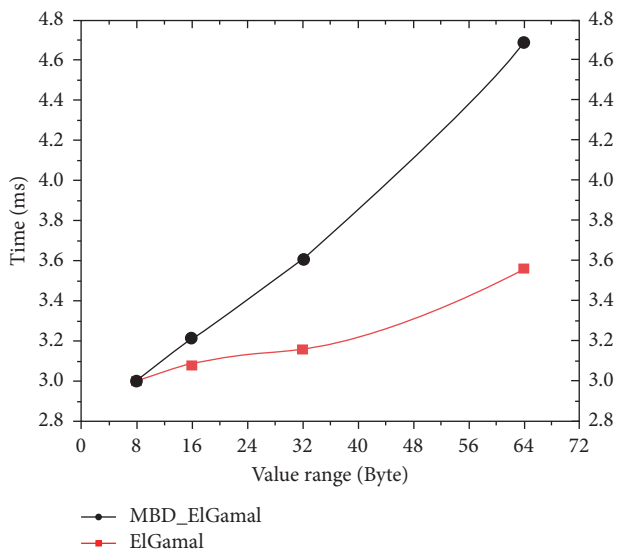


FIGURE 3: Decryption time.

TABLE 4: Encryption scheme comparison.

Schemes	Setup time	Enc time	Dec time
This paper	52 s	5.4 ms	3.4 ms
Paillier	403 ms	27 ms	7 ms

of ElGamal based on multibase decomposition can be increased by more than 1.2 times. The decryption time can be increased by more than 1.3 times.

For transaction data privacy, we compare this scheme with the Paillier encryption scheme with the same security level. As shown in Table 4, we only compare the encrypted and decrypted parts. It can be seen from the figure that our scheme is about 5 times and 2 times higher than Paillier’s encryption efficiency. This is because our solution requires a longer initialization time and sacrifices the time overhead of initial parameters in exchange for more efficient encryption and decryption time.

7. Conclusion

In this paper, we have designed a supervisable transaction data privacy protection scheme, which settles the problem of transaction privacy and effective supervision in the blockchain-based energy transaction scheme. Specifically, we combine Pedersen commitment and zero-knowledge proof technology to ensure the authenticity of transaction data, which effectively prevents malicious users from using random amounts for encryption to defraud the regulator. Simultaneously, the transaction data can be verified in a ciphertext environment. In the process of generating audit tickets, we introduced the ElGamal encryption and decryption method based on multibase decomposition to improve the efficiency of encryption and decryption. And the Bulletproofs range proof technology is introduced in the transaction creation process, which improves the efficiency of transaction verification. Security and performance analysis show that the scheme can audit a certain transaction or multiple transactions in the ledger and effectively protect the privacy of transaction amounts.

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 paper was supported by the Natural Science Foundation under Grant nos. 61962009 and 62162008; Major Scientific and Technological Special Project of Guizhou Province under Grant nos. 20183001 and 20193003; Science and Technology Support Plan of Guizhou Province ((2020) 2Y011); Foundation of Guangxi Key Laboratory of Cryptography and Information Security (GCIS202118); and Shandong Provincial Natural Science Foundation (ZR202103050289).

References

- [1] C. S. Wright, "Bitcoin: a peer-to-peer electronic cash system," *SSRN Electronic Journal*, no. 9, 2008.
- [2] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
- [3] A. Singh, R. M. Parizi, Q. Zhang, K.-K. R. Choo, and A. Dehghantaha, "Blockchain smart contracts formalization: approaches and challenges to address vulnerabilities," *Computers & Security*, vol. 88, Article ID 101654, 2020.
- [4] V. Aleksieva, H. Valchanov, and A. Huliyan, "Application of smart contracts based on ethereum blockchain for the purpose of insurance services," in *Proceedings of the 2019 International Conference on Biomedical Innovations and Applications (BIA)*, pp. 1–4, IEEE, Varna, Bulgaria, November 2019.
- [5] T. Li, Y. Chen, Y. Wang et al., "Rational protocols and attacks in blockchain system," *Security and Communication Networks*, vol. 2020, Article ID 8839047, 11 pages, 2020.
- [6] Y. Wang, G. Yang, T. Li et al., "Optimal mixed block withholding attacks based on reinforcement learning," *International Journal of Intelligent Systems*, vol. 35, no. 12, pp. 2032–2048, 2020.
- [7] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "Psspr: a source location privacy protection scheme based on sector phantom routing in wsns," *International Journal of Intelligent Systems*, 2021.
- [8] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, 2021.
- [9] G. Yang, Y. Wang, Z. Wang, Y. Tian, X. Yu, and S. Li, "Ipbms: an optimal bribery selfish mining in the presence of intelligent and pure attackers," *International Journal of Intelligent Systems*, vol. 35, no. 11, pp. 1735–1748, 2020.
- [10] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: anonymity for bitcoin with accountable mixes," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 486–504, Springer, Christ Church, Barbado, March 2014.
- [11] X. Yu, Z. Wang, Y. Wang et al., "Impsuic: a quality updating rule in mixing coins with maximum utilities," *International Journal of Intelligent Systems*, vol. 36, no. 3, pp. 1182–1198, 2021.
- [12] S. Noether, "Ring signature confidential transactions for monero," *IACR Cryptol. ePrint Arch*, vol. 1, p. 1098, 2015.
- [13] Y. Li, G. Yang, W. Susilo, Y. Yu, M. H. Au, and D. Liu, "Traceable monero: anonymous cryptocurrency with enhanced accountability," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 679–691.
- [14] E. B. Sasson, A. Chiesa, C. Garman et al., "Zerocash: decentralized anonymous payments from bitcoin," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, pp. 459–474, IEEE, Berkeley, CA, USA, May 2014.
- [15] G. Fuchsbaauer, M. Orrù, and Y. Seurin, "Aggregate cash systems: a cryptographic investigation of mumblewimble," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 657–689, Springer, Darmstadt, Germany, May 2019.
- [16] G. Betarte, M. Cristiá, C. Luna, A. Silveira, and D. Zanarini, "Towards a formally verified implementation of the mumblewimble cryptocurrency protocol," in *Proceedings of the International Conference on Applied Cryptography and Network Security*, pp. 3–23, Springer, Rome, Italy, October 2020.
- [17] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, "A blockchain-based machine learning framework for edge services in iiot," *IEEE Transactions on Industrial Informatics* 1 page, 2021.
- [18] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in dwsns," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.
- [19] Chen, Y. Dong, S. Li, T. Wang, Y. Zhou, and Huiyu, "Dynamic multi-key FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2021.
- [20] K. Wüst, K. Kostianen, V. Čapkun, and S. Čapkun, "Prcash: fast, private and regulated transactions for digital currencies," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 158–178, Springer, Frigate Bay, St. Kitts and Nevis, February 2019.
- [21] N. Narula, W. Vasquez, and M. Virza, "zkledger: privacy-preserving auditing for distributed ledgers," in *Proceedings of the 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pp. 65–80, Renton, WA, USA, April 2018.
- [22] H. Kang, T. Dai, N. Jean-Louis, S. Tao, and X. Gu, "Fabzk: supporting privacy-preserving, auditable smart contracts in hyperledger fabric," in *Proceedings of the 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 543–555, IEEE, Portland, OR, USA, June 2019.
- [23] M. Blum, A. De Santis, S. Micali, and G. Persiano, "Noninteractive zero-knowledge," *SIAM Journal on Computing*, vol. 20, no. 6, pp. 1084–1118, 1991.
- [24] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: short proofs for confidential transactions and more," in *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP)*, pp. 315–334, IEEE, San Francisco, CA, USA, May 2018.
- [25] J. Groth and M. Maller, "Snarky signatures: minimal signatures of knowledge from simulation-extractable snarks," in *Proceedings of the Annual International Cryptology*

Conference, pp. 581–612, Springer, Santa Barbara, CA, USA, August 2017.

- [26] A. Fiat and A. Shamir, “How to prove yourself: practical solutions to identification and signature problems,” in *Proceedings of the Conference on the Theory and Application of Cryptographic Techniques*, pp. 186–194, Springer, Santa Barbara, CA, USA, August 1986.
- [27] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

Research Article

Lodestone: An Efficient Byzantine Fault-Tolerant Protocol in Consortium Blockchains

Chen Shan  and Lei Fan 

School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai 201100, China

Correspondence should be addressed to Lei Fan; fanlei@sjtu.edu.cn

Received 29 October 2021; Accepted 16 November 2021; Published 3 December 2021

Academic Editor: Yuling Chen

Copyright © 2021 Chen Shan and Lei Fan. 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.

We present Lodestone, a chain-based Byzantine fault-tolerant (BFT) state machine replication (SMR) protocol under partial synchrony. Lodestone enables replicas to achieve consensus with two phases of voting and enjoys (1) optimistic responsiveness and (2) linear communication complexity on average. Similar to the state-of-the-art chain-based BFT protocols, Lodestone can be optimized with a pipelining idea elegantly. We implement pipelined Lodestone and deploy experiments to evaluate its performance. The evaluation results demonstrate that Lodestone has a lower latency than HotStuff under various workloads.

1. Introduction

Consensus in blockchain systems, known as state machine replication (SMR), has attracted more and more interest in recent years. When focusing on permissioned blockchains on so-called consortium blockchains, chain-based Byzantine fault-tolerant (BFT) SMR protocols [1–8] under partial synchrony have been widely used to achieve consistency. In general, chain-based BFT SMR protocols follow the conventional propose-vote paradigm where there exists a special role often called leader who is responsible for packing clients' requests into proposals, and then all players achieve consensus on these proposals via multiple (two or three) phases of voting.

PBFT [9], as the first practical BFT SMR protocol under the partial synchronous network [10], achieves safety even under the asynchronous network and liveness when the network gets synchronous. However, the view-change subprotocol in PBFT, with an $O(n)^3$ communication complexity, is too heavy to be practical. Tendermint [6] innovatively employs a lock-commit scheme, similar to the paradigm in [11], that a replica should lock on the proposal

he has voted COMMIT for. This allows one replica to decide if voting for one proposal according to his own local states and the leader has no need to prove the safety of his proposal. Casper [7, 8] takes a similar strategy and also implies a pipelining idea for a further improvement. However, both Tendermint and Casper sacrifice optimistic responsiveness in that there needs to be a fixed interval between proposals to guarantee liveness since a new leader has to ensure that he has observed all other nonfaulty replicas' lock state; otherwise, his new proposal may not be accepted. HotStuff [1] creatively introduces another phase of vote to achieve both linear view-change and optimistic responsiveness. The additional phase guarantees that a new leader can construct this proposal safely only with $n - f$ replicas' states. However, in pipelined HotStuff, the three-phase voting scheme not only brings about an increase in latency but also causes an implicit liveness problem. In pipelined HotStuff, there needs to be four consecutive nonfaulty leaders to make a decision which cannot be guaranteed in the $n = 3f + 1$ setting. This means pipelined HotStuff cannot provide liveness in the worst case even under the crash fault-tolerant model, which has also been discussed as the silence attack [12].

We present Lodestone, a novel chain-based BFT SMR protocol, which achieves the following combined properties under partial synchrony:

- (1) Two-phase voting with optimistic responsiveness: Lodestone can achieve responsiveness when liveness is guaranteed after GST. That is, the total time of confirmation on one honest leader’s proposal only relies on the actual network delay instead of any apriori upper bound assumption of network delay.
- (2) Linear average-case communication complexity: a view-change subprocess in Lodestone costs $O(n)$ message complexity on average and $O(n^2)$ in the worst case. Here, we follow the measurement of message complexity in HotStuff [1] which counts the total number of authenticators received by one player in the protocol to achieve a decision.
- (3) Deterministic liveness under static corruption: pipelined Lodestone can also achieve deterministic liveness under static corruption where leaders are rotated in a round-robin manner.

The difference in detail between these protocols can be shown in figures, of which Figures 1 and 2 show HotStuff and existing two-phase voting protocols, respectively, while Figure 3 demonstrates our solution.

2. Other Related Works

2.1. BFT SMR Protocols in Alternative Assumptions. There are also many works considering about BFT protocols in alternative assumptions.

Firstly, about the network assumption, many recent protocols under a synchronous network [13–15] or under an asynchronous network [16–18] make efforts to reduce confirmation latency and achieve practical throughput. Secondly, about the corruption assumption, some recent works [19, 20] are aimed to provide higher assurance on blocks even if the adversary corrupts more than f replicas in the future. In this paper, we only concentrate on static security under a partially synchronous network.

2.2. Single-Shot BFT Protocols. There are some related works [21–23] focusing on the single-shot BFT problem which explore the optimal latency bound when the leader is nonfaulty under various resilience assumptions. Though it is still a long way to construct a protocol from single shot to multishots, some results are interesting and may be combined with our protocols in the future work.

2.3. View Synchronization. Another related line of work is about view synchronization [24–26] in the partial synchrony setting. It is also necessary in Lodestone for nonfaulty replicas to stay in the same view for a sufficient long time, and then liveness can be guaranteed. However, view synchronization is not the key point of Lodestone, and we assume that replicas are able to stay in the same view for a sufficient long time after GST.

3. Materials and Methods

3.1. Models

- (1) Threat model: we consider a permissioned system consisting of n replicas, indexed by $i \in [n]$, where $[n] = \{1, 2, \dots, n\}$. We assume a polynomially bounded adversary who can corrupt less than $n/3$ replicas. The replicas corrupted by the adversary can deviate from the prescribed protocol arbitrarily within their capabilities which are also called Byzantine faulty replicas, while the remaining ones are nonfaulty. We only consider a static corruption model in that the adversary chooses which replicas to corrupt prior to the execution.
- (2) Network model: we assume that each pair of replicas is connected by a reliable authenticated point-to-point channel. Messages are propagated through a partially synchronous network [10] in that there is an unknown global stabilization time (GST). After GST, a message sent by a nonfaulty replica will be delivered to all nonfaulty replicas with a known bound Δ , though the delivery schedule is determined by the adversary.
- (3) Cryptographic primitives: we assume a cryptographic hash function $\text{Hash}(m)$ and a standard digital signature scheme. We also assume a (t, n) threshold signature scheme [27, 28] which provides the following interfaces:

$\text{ThresholdSetup}(1^\lambda)$ generates a pair of key shares $\{pk_i, sk_i\}$ for replica i along with a global public key PK.

$\text{ThresholdSign}_i(m)$ produces a signature share λ_i of message m with sk_i .

$\text{ThresholdVerifyShare}(m, i, \lambda_i)$ verifies if λ_i is a valid signature share of message m .

$\text{ThresholdCombine}(m, i, \lambda_{i \in I})$ produces the threshold signature λ of message m from t signature shares where $I \subset [n]$ and $|I| = t$.

$\text{ThresholdVerify}(m, \lambda)$ verifies if λ is a valid threshold signature of message m with PK.

We use an $(n-f, n)$ threshold signature scheme in our following protocols which is assumed to provide robustness and nonforgeability.

- (4) Problem definition: we now give the definition of a chain-based BFT SMR protocol. Each replica in a chain-based BFT SMR protocol receives requests from clients and maintains a sequence of blocks called a blockchain. Blocks in a blockchain are chained by hash digest, and thus, each block in a blockchain has its own position denoted as its height. Given a blockchain C and a block $b \in C$, all blocks in C lower than b are ancestor blocks of b , and all blocks in C higher than b are descendant blocks of b . Two blocks b_1 and b_2 are conflicting if and only if b_1 is neither an ancestor nor a descendant block of b_2 . Each block includes a batch of requests, and one

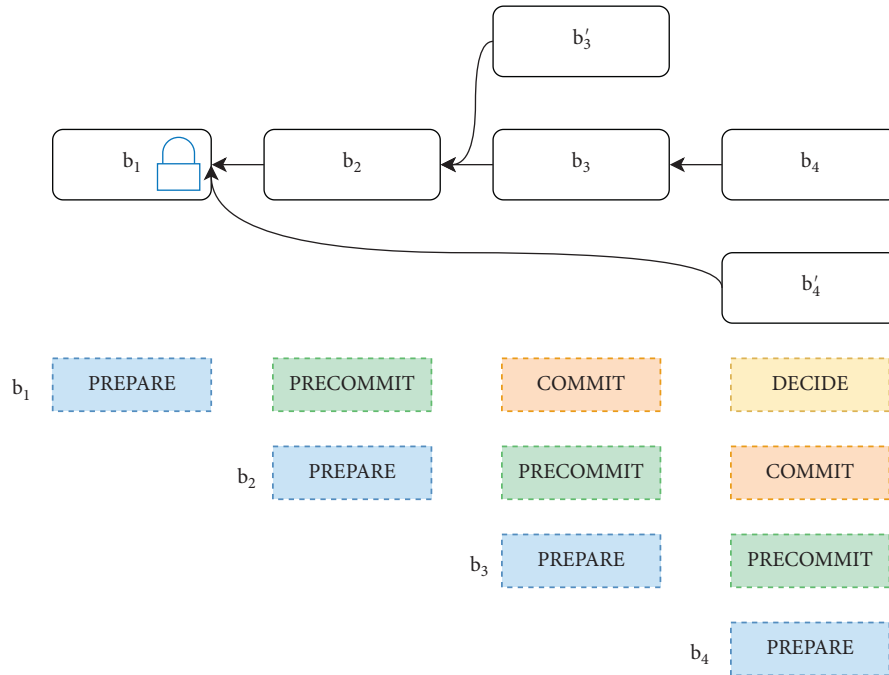


FIGURE 1: Pipelined HotStuff. $b_1, b_2, b_3,$ and b_4 are in the normal case, while $b_1, b_2, b_3',$ and b_4' are in the timeout case. In the timeout case, nonfaulty replicas who have received b_3' will lock on b_1 but will still vote for b_4' since b_4' also extends from b_1 .

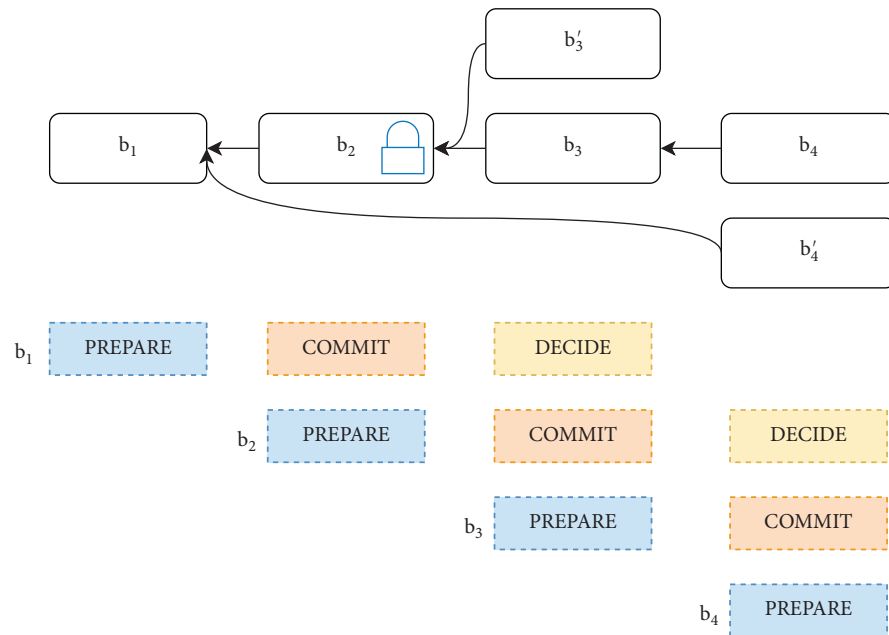


FIGURE 2: Casper or the two-chain variant of pipelined HotStuff. $b_1, b_2, b_3,$ and b_4 are in the normal case, while $b_1, b_2, b_3',$ and b_4' are in the timeout case. In the timeout case, the leader of b_4' did not receive b_3' in time, while some nonfaulty replicas received b_3' and have locked on b_2 . Therefore, they would not vote for b_4' since b_4' conflicts with b_2 .

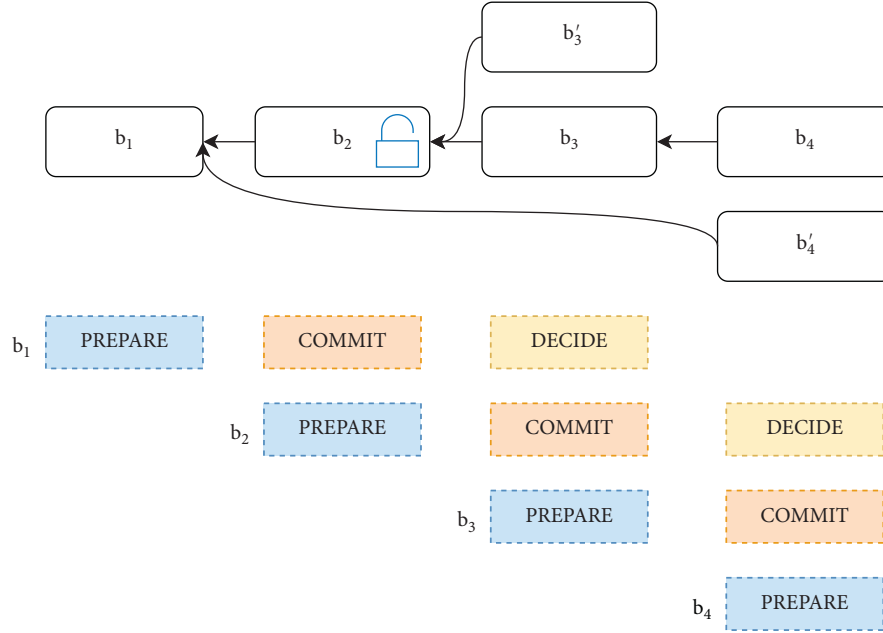


FIGURE 3: Pipelined Lodestone. $b_1, b_2, b_3,$ and b_4 are in the normal case, while $b_1, b_2, b'_3,$ and b'_4 are in the timeout case. In the timeout case, the leader of b'_4 will attach a proof to his proposal which allows nonfaulty replicas who have locked on b'_2 to get unlocked and vote for b'_4 .

replica's blockchain consists of all blocks which have been finalized. Replicas execute requests in finalized blocks in the sequence of the blockchain and then respond to clients. For simplicity, we do not model clients and assume all requests are sent to all replicas. Thus, we say requests from clients are input to all replicas, and at any time, the output of one replica is his blockchain.

A secure chain-based BFT SMR protocol should satisfy both safety and liveness defined below (in the presence of an arbitrary adversary with all but negligible probability).

Definition 1. (safety). At any time, if two nonfaulty replicas each have their blockchains denoted as C_1 and C_2 , then it must be either $C_1 \preceq C_2$ or $C_2 \preceq C_1$, where \preceq means "is a prefix of or equal to." In other words, two nonfaulty replicas will never finalize different blocks at the same height.

Definition 2. (liveness). If a request req has been inputted to all replicas at time $t > T_{\text{start}}$, then at time $t + T_{\text{confirm}}$, any nonfaulty replica must output a blockchain which includes req , where T_{start} is the time after which the protocol provides liveness and T_{confirm} is a bounded constant.

3.2. Preliminaries

- (1) Block: we now format a block b in our protocol as $b = \text{parent, view, txs, and hash}$ where parent is the hash digest of the parent block of b , view is the view in which b is proposed, txs is a batch of requests from clients, and hash is the hash digest of b (i.e., $b.\text{hash} = \text{Hash}(b.\text{parent}, b.\text{view}, b.\text{txs})$). In Lodestone, there are three states for one block, namely,

PROPOSED, PREPARED, and COMMITTED. Once a block reaches the COMMITTED state, then the block itself and its all ancestor blocks are finalized and can be executed sequentially.

- (2) Local states: replicas in Lodestone need to maintain two local states for the protocol execution. The first denoted as currView represents the current view number and implies the current leader. It is noted that using the round-robin manner to rotate leaders guarantees that (i) the protocol will be greeted with a nonfaulty leader after at most f consecutive views; (ii) there exist three consecutive views of which leaders are all nonfaulty which will be proved later. The second is lockedQC which stores qc with the largest view number one replica has voted commit for. Once a nonfaulty replica locks on qc (i.e., sets lockedQC as qc), he will only vote for blocks extending from block b that $\text{Hash}(b) = qc.\text{hash}$ unless he ensures the majority of nonfaulty replicas have turned to a conflicting branch before locking on qc .
- (3) Promise and promise-set: during a view-change, each replica additionally sends to the leader a *promise*, evidence that helps the leader locate the highest COMMITTED block and prove that he does so in an honest manner. A *promise* for view v indicates that the replica has not voted commit at view v . If $n - f$ replicas send *promises* for the view v , then there is no block which has a larger view number than v could be COMMITTED. When the leader proposes a block with the $n - f$ *promises*, all the nonfaulty replicas can accept it safely. A *promise* from replica P includes a tuple v, \hat{v} and the corresponding signature share $\sigma = \text{ThresholdSign}(v, \hat{v})$

from P . We denote $p = v, \hat{v}, \sigma$ for simplicity. We say p from P for the current view \hat{v} represents that he did not vote commit when he was in the view v . Let \tilde{v} be the last view in which P has voted commit. P will generate a set of *promises* for each view from $\tilde{v} + 1$ to *currView*. We denote it as a *promise-set*.

- (4) NullQC: before entering a new view, each nonfaulty replica P_i will send his promise-set *promise-set* to the new leader, along with his *lockedQC*. Upon receiving valid promise-sets from $n - f$ distinct replicas, the new leader will select *lockedQC* with the largest view number as *highQC* and propose his new block following the block b where $\text{Hash}(b) = \text{highQC.hash}$. Let I be the set of indexes of the $n - f$ replicas; we have $|I| = n - f$. We use \hat{v} to denote *currView* and \tilde{v} to denote *highQC.view*. Each promise-set must contain a *promise* that $\text{promise}_i^{v, (v+1)} = \tilde{v} + 1, \hat{v}, \lambda_i^{v, (v+1)}$ where $i \in I$. The new leader can combine these $n - f$ signature shares as a threshold signature to prove that *highQC* he selected is exactly the one with the largest view number among $n - f$ replicas. This extra proof is denoted as *nullQC* in Lodestone. We have $\text{nullQC} = \text{ThresholdCombine}(\tilde{v} + 1, \hat{v}, \{\lambda_i^{v, (\tilde{v}+1)}\})$, where $i \in I$.

3.3. *Pipelined Lodestone*. We are now ready to describe our protocol pipelined Lodestone. We first define some utilities as shown in Figure 4.

Then, we formalize pipelined Lodestone with algorithms. The protocol runs in a succession of views denoted as *currView*. Each view number is mapped into a leader in a round-robin manner. The leader will execute Algorithm 1, all replicas will execute Algorithm 2, and then the next leader will execute Algorithm 3. When timeout triggers during any wait-for procedure in one replica's local view, he will execute Algorithm 4. We omit any check for brevity.

4. Safety, Liveness, and Communication Complexity

4.1. Safety

Lemma 1. *For any two valid quorum certificates qc_1 and qc_2 , if $qc_1 \cdot \text{view} = qc_2 \cdot \text{view}$, then $qc_1 \cdot \text{block} = qc_2 \cdot \text{block}$.*

Proof. For any valid qc , at least $n - 2f$ nonfaulty replicas have sent their relevant signature shares, namely, at least $n - 2f$ nonfaulty replicas have voted for the block which $qc \cdot \text{block}$ represents in $qc \cdot \text{view}$. Suppose qc_1 and qc_2 are two valid qc such that $qc_1 \cdot \text{view} = qc_2 \cdot \text{view}$, but $qc_1 \cdot \text{block} \neq qc_2 \cdot \text{block}$. We must have that at least $n - 2f$ nonfaulty replicas voted for the $qc_1 \cdot \text{block}$ and also at least $n - 2f$ nonfaulty replicas voted for the $qc_2 \cdot \text{block}$ in the same view. Thus, the intersection of the two sets at least includes one nonfaulty replica since $2 \times (n - 2f) > n - f$, a contradiction to Algorithm 2 that a nonfaulty replica can only vote once in a view.

Theorem 1. *In pipelined Lodestone, two conflicting blocks cannot be both COMMITTED.*

Proof. Let b_1 and b_2 be two conflicting blocks which are both COMMITTED, namely, there exists a two-chain $\langle b_{-1}, b_{-1}', b_{-1}'' \rangle$ and also one in the form of $b_{-2}, b_{-2}', b_{-2}''$. W.l.o.g, we can assume that $b_2 \cdot \text{view} > b_1' \cdot \text{view}$ with Lemma 1. Since b_1 is COMMITTED, at least $n - 2f$ nonfaulty replicas have voted for b_{-1}' and then locked on b_1 . Let \hat{b} be the PREPARED block with the smallest view number that satisfies $\hat{b} \cdot \text{view} > b_1' \cdot \text{view}$, and \hat{b} conflicts with b_1 . Such \hat{b} must exist since b_2 satisfies all these conditions. Now, consider the proposal m of \hat{b} . At least $n - 2f$ nonfaulty replicas have locked on b_1 in $b_1' \cdot \text{view}$. These $n - 2f$ nonfaulty replicas will only promise on view number larger than $b_1 \cdot \text{view}$ since then. Therefore, $m \cdot \text{proof} \cdot \text{view} > b_1 \cdot \text{view}$. And due to the minimality of \hat{b} , we must have $m \cdot \text{qc.view} < b_1 \cdot \text{view}$. Then, $m \cdot \text{proof.view} > m \cdot \text{qc.view} + 1$, a contradiction to the validity of *nullQC*.

4.2. Liveness

Lemma 2. *Under an $n = 3f + 1$ setting where leaders are rotated in a round-robin manner, in any consecutive $n + 2$ views, there are three consecutive views all led by non-faulty replicas.*

Proof. Consider any n consecutive views denoted as $v, v + 1, v + 2, \dots, v + n - 1$. There are three cases:

- (i) The leader of v is faulty. Then, for the view from $v + 1$ to $v + n - 1$, there are $f - 1$ faulty leaders while $2f + 1$ nonfaulty leaders. The $f - 1$ faulty leaders can at most separate $2f + 1$ nonfaulty leaders into f segments, with at least one segment owning 3 nonfaulty leaders.
- (ii) The leader of v is nonfaulty, while the leader of $v + 1$ is faulty. Then, for the view from $v + 2$ to $v + n$, there are $f - 1$ faulty leaders while $2f + 1$ nonfaulty leaders. Then, the situation becomes the same as the first case.
- (iii) Both v and $v + 1$ are led by nonfaulty leaders. Then, if the leader of $v + 2$ is nonfaulty, three consecutive nonfaulty leaders exist. Otherwise, for the view from $v + 3$ to $v + n + 1$, there are $f - 1$ faulty leaders while $2f + 1$ nonfaulty leaders. Then, the situation becomes the same as the first case.

Lemma 3. *If a nonfaulty leader of view v collects $n - f$ valid view-change messages, he can always propose his new block, along with valid *nullQC* that $\text{nullQC} \cdot \text{view} = \text{highQC} \cdot \text{view} + 1$ where *highQC* points to the parent block of his new block.*

Proof. Let P_i be the nonfaulty leader of view v . If P_i collects $n - f$ valid view-change messages, he will select qc with the largest view number among them as *highQC*. Each of the $n - f$ valid view-change messages must also include a promise on $\text{highQC.view} + 1$; then, P_i can combine a threshold signature with these $n - f$ signature shares and construct *nullQC* that $\text{nullQC} \cdot \text{view} = \text{highQC} \cdot \text{view} + 1$.

```

function VOTE (view,hash,id)
  v.id=i d
  v.view=view
  v.hash=hash
  v.sigShare=ThresholdSigni (v)
  return v
end function

function MSG (block, qc, proof, view)
  m.block=block
  m.qc=qc
  m.proof=proof
  m.view=view
  return m
end function

function BLOCK (view,parent)
  b.view=view
  b.requests=requests
  b.parent=parent
  b.hash= Hash (b)
  return b
end function

function QC (view, hash, V)
  qc.view=view
  qc.block=hash
  qc.proof=ThresholdCombine (qc, {⟨v. id, v. sigShare⟩ | v ∈ V})
  return qc
end function

function NULLQC (view, currView, T)
  nullQC.view=view
   $\Sigma = \{ \langle i, t, \sigma_i^{view, currView} \rangle | \forall T_i \in T, t \in T_i \}$ 
  nullQC.proof=ThresholdCombine (⟨view, currView⟩  $\Sigma$ )
  return nullQC
end function

function PROOF (startView, currView, id)
  T=∅
  for v=startView to currView do
     $\sigma_i^{v,\hat{v}} = \text{ThresholdSign}_i(\langle v,\hat{v} \rangle)$ 
    T = T ∪ {⟨⟨v,  $\hat{v}$ ⟩,  $\sigma_i^{v,\hat{v}}$ ⟩}
  end for
  return T
end function

```

FIGURE 4: Utilities in the pipelined Lodestone protocol.

- (1) Wait for $n - f$ VIEW-CHANGE message $m \in M$ of currView-1
- (2) $highQC \leftarrow \text{argmax}_{m \in M} \{m.qc.view\}$
- (3) if $lockedQC.view < highQC.view$
- (4) $lockedQC \leftarrow highQC$
- (5) $\hat{T} \leftarrow \{m.proof | m \in \}$
- (6) $nullQC \leftarrow \text{NullQC}(highQC.view + 1, currView, \hat{T})$
- (7) $b \leftarrow \text{BLOCK}(currView, lockedQC.hash)$
- (8) Broadcast MSG (PROPOSE, b, lockedQC, NullQC)

ALGORITHM 1: Pipelined Lodestone protocol: as the leader of currView.

```

(1) Wait for RROPOSE message  $m$  from LEADER ( $currView$ )
(2)  $b \leftarrow m.block$ 
(3) if  $m.qc.view + 1 = m.proof.view$ .
(4)    $v \leftarrow VOTE(currView, b, hash, i)$ 
(5)   send MSG (GENERIC,  $b, hash, \perp, v$ ) to LEADER ( $currView + 1$ ).
(6) if  $m.qc.view > lockedQC.view$ 
(7)    $lockedQC \leftarrow m.qc$ 
(8)    $b' \leftarrow b.parent, b'' \leftarrow b'.parent$ 
(9)   if  $b''.view + 1 = b'.parent$ 
(10)  DECIDE ( $b''$ )

```

ALGORITHM 2: Pipelined Lodestone protocol: as a replica P_i of $currView$.

```

(1) Wait for  $n - f$  matched GENERIC messages  $\tilde{M}$  of  $currView$ 
(2)  $hash \leftarrow \tilde{m}.block, \forall \tilde{m} \in \tilde{M}$ 
(3)  $\tilde{V} \leftarrow \{\tilde{m}.proof \mid \tilde{m} \in \tilde{M}\}$ 
(4)  $lockedQC \leftarrow QC(currView, hash, \tilde{V})$ 

```

ALGORITHM 3: Pipelined Lodestone protocol: as the next leader of $currView$.

```

Jump here if TIMEOUT triggers during any wait procedure or before entering  $currView+1$ 
(1)  $T \leftarrow PROOF(lockedQC.view + 1, currView + 1, i)$ 
(2) send MSG (VIEW-CHANGE,  $\perp, lockedQC, T$ ) to LEADER ( $currView + 1$ )

```

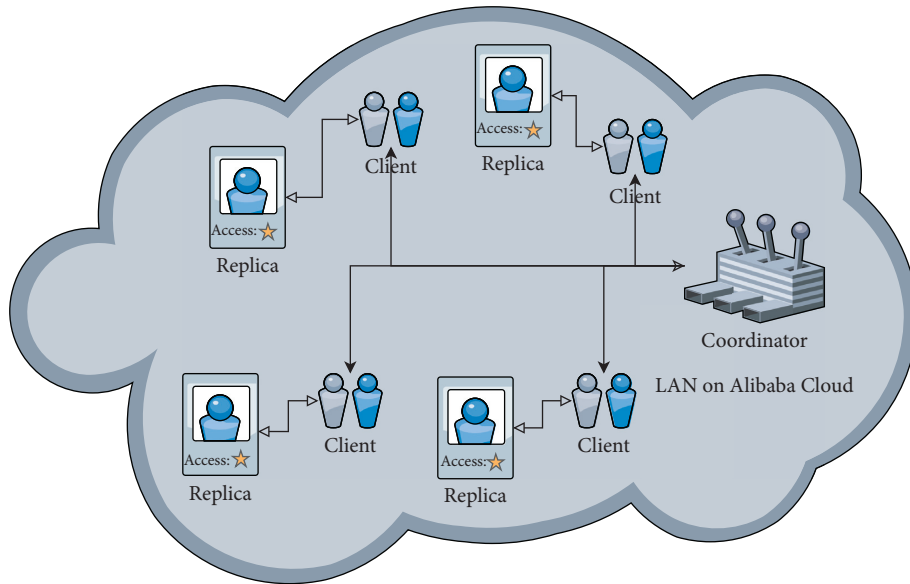
ALGORITHM 4: Pipelined Lodestone protocol: TimeOut $currView$.

FIGURE 5: The architecture diagram of experiments.

Theorem 2. In pipelined Lodestone, after GST, any request from clients will be included in the finalized block in a bounded time.

Proof. According to Lemma 2, there are three consecutive views led by nonfaulty replicas in any consecutive $n + 2$ views, denoted as $v, v + 1, v + 2$. There exists a bounded time

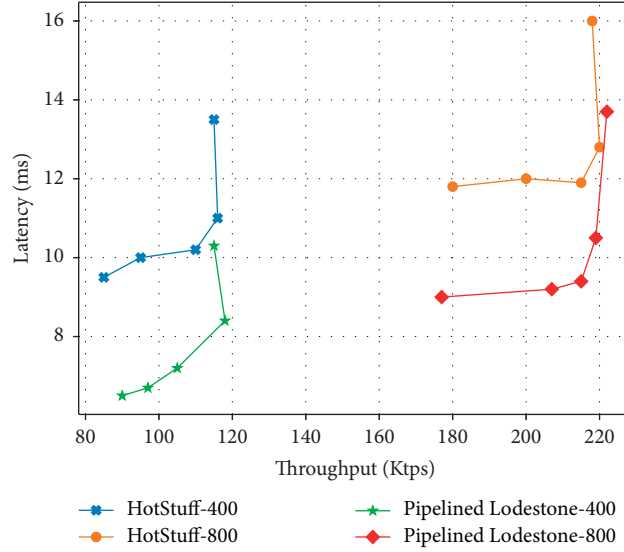


FIGURE 6: Throughput vs. latency with different choices of batch size, 10 replicas, and zero-sized payload.

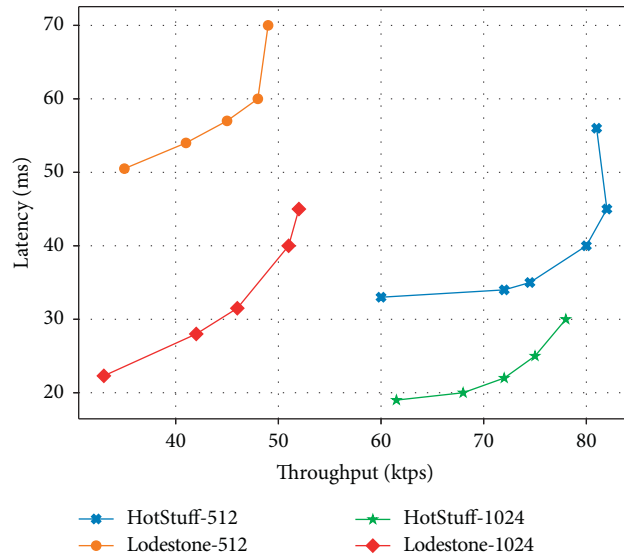


FIGURE 7: Throughput vs. latency with different choices of payload size, 10 replicas, and batch size of 800.

duration T that if all nonfaulty replicas stay in the same view during T , then the leader can propose his new block with $\text{nullQC} \cdot \text{view} = \text{highQC} \cdot \text{view} + 1$ according to Lemma 3. Thus, the first nonfaulty leader of v will propose a block b , and all nonfaulty replicas will vote for b . Then, the second nonfaulty leader of $v+1$ will collect $n-f$ votes for b and propose a block b' with $\text{highQC}' \cdot \text{view} = v \wedge \text{nullQC}' \cdot \text{view} = v+1$. All nonfaulty replicas will also vote for b' . Then, the third nonfaulty leader of $v+2$ will collect $n-f$ votes for b' and propose a block b'' with $\text{highQC}'' \cdot \text{view} = v+1$. Therefore, b , b' , and b'' form a two-chain b, b', b'' , and all nonfaulty replicas will consider b as COMMITTED after receiving b'' . For any time t , the three

consecutive views led by nonfaulty players will come within a bounded time duration after t . Therefore, there exists a bounded constant T_{confirm} in that for any request input to all replicas at time t , it will be included in the finalized block at $t + T_{\text{confirm}}$.

4.3. Communication Complexity. We now discuss the communication complexity of pipelined Lodestone. It is noted that we only consider the communication overhead when liveness can be guaranteed after GST. When the network is asynchronous, there may be unbounded views without making a decision. In fact, we can allow replicas at

most send f promises during a view-change which has no effect on the safety property since it is impossible to guarantee liveness under an asynchronous network [29].

Theorem 3. *After GST, pipelined Lodestone achieves a linear communication complexity on average while $O(n^2)$ in the worst case.*

Proof. When liveness can be guaranteed after GST, whenever two consecutive views are led by nonfaulty replicas, the second leader can generate qc , and all nonfaulty replicas will update their *lockedQC*. Therefore, in the worst case, there are $2f + 1$ consecutive views without generating new qc . And then, one replica needs to send $2f + 1$ promises on these $2f + 1$ views. And thus, the total complexity of a view-change is $O(n^2)$ in the worst case.

Now, we discuss the communication complexity of a view-change in the average case. The probability of one view led by a faulty replica can be considered as $1/3$ while $2/3$ by a nonfaulty leader approximately with the assumption $n = 3f + 1$. Given any view v , let X be the length of views before v without any two consecutive views led by nonfaulty replicas. Let $P(X)$ be the probability distribution of X . For any view v , $X = k$ occurs when

- (i) Either the leader of $v - 1$ is faulty and $X = k - 1$ holds for $v - 1$
- (ii) Or the leader of $v - 1$ is nonfaulty while the leader of $v - 2$ is faulty and $X = k - 2$ holds for $v - 2$

As the assumption, $v - 1$ and $v - 2$ also follow the same probability distribution which is denoted as $P(X = k - 1)$ and $P(X = k - 2)$, respectively. Then, $P(X = k)$ can be expressed as

$$P(X = k) = \frac{1}{3}P(X = k - 1) + \frac{2}{9}P(X = k - 2). \quad (1)$$

The expectation $E(X)$ is the length of views before v without any two consecutive views led by nonfaulty players on average. $E(k)$ is also the total number of signatures in the promise-set of one replica on average. We have $E(X) = 7/4$ after simplification. Therefore, the expectation of the total communication overhead of a view-change is $O(n)$ for all n replicas in a view.

5. Results and Discussion

5.1. Implementation and Setup. We have implemented both pipelined Lodestone and pipelined HotStuff in C++ language with the same codebase for a fair comparison, taking the implementation (<https://github.com/hot-stuff/libhotstuff>) in HotStuff’s paper [1] as a reference. We use Ed25519 for common digital signatures and BLS threshold signatures (<https://github.com/herumi/bls>) for combining signatures in our protocol.

We deploy our experiments on Alibaba Cloud using ecs.ic5.4xlarge instances. The round-trip delay between two instances is less than 1 millisecond, with the bandwidth about 5 Gbps.

In all experiments, besides all players and clients, we develop a coordinator who is responsible for notifying all clients sending requests to players. The coordinator collects measurement data to compute the throughput and end-to-end latency of clients. Figure 5 shows the architecture in our experiments.

We first evaluate these two protocols with zero-sized payload and different choices of batch sizes to get rid of the effects of payload size. Figure 6 shows that Lodestone has a prominent lower latency compared with HotStuff under the batch size of both 400 and 800 benefiting from conserving one phase of the vote.

Figure 7 depicts different payload sizes for both systems as 512 bytes and 1024 bytes, with a fixed batch size of 800. In such settings, Lodestone still enjoys a remarkable lower latency compared with HotStuff and provides comparable throughput.

6. Conclusions

We presented pipelined Lodestone, a chain-based BFT SMR protocol, which achieves linear view-change on average and optimistic responsiveness with only two phases of voting. Through the experimental results, pipelined Lodestone provides a lower latency and comparable throughput compared with HotStuff in various workloads and network scales.

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 there are no conflicts of interest regarding the publication of this paper.

References

- [1] M. Yin, D. Malkhi, K. Michael, M. K. Reiter, and G. G. Gueta, “HotStuff: BFT consensus with linearity and responsiveness,” in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pp. 347–356, Toronto ON Canada, July 2019.
- [2] I. Abraham, D. Malkhi, K. Nayak, L. Ren, and M. Yin, “Sync HotStuff: synchronous SMR with 2Δ latency and optimistic responsiveness,” *IACR Cryptol. ePrint Arch*, vol. 2019, 270 pages, 2019.
- [3] T. H. C. Hubert, R. Pass, and E. Shi, “PiLi: an extremely simple synchronous blockchain,” *IACR Cryptol. ePrint Arch*, vol. 2018, 980 pages, 2018.
- [4] T. H. C. Hubert, R. Pass, and E. Shi, “PaLa: a simple partially synchronous blockchain,” *IACR Cryptol. ePrint Arch*, vol. 2018, 981 pages, 2018.
- [5] B. Y. Chan and E. Shi, “Streamlet: textbook streamlined blockchains,” in *Proceedings of the second ACM Conference on Advances in Financial Technologies*, pp. 1–11, New York, NY, USA, October 2020.
- [6] E. Buchman, J. Kwon, and Z. Milosevic, “The latest gossip on BFT consensus,” 2018, <https://dblp.org/rec/journals/corr/abs-1807-00734.html?view=bibtex>, Article ID 04938.

- [7] V. Buterin and V. Griffith, “Casper the Friendly Finality Gadget,” 2017, <https://arxiv.org/abs/1710.09437>, Article ID 09437.
- [8] V. Buterin, D. Reijnders, S. Leonardos, and G. Piliouras, “Incentives in ethereum’s hybrid casper protocol,” in *Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 236–244, Seoul, May 2019.
- [9] M. Castro and B. Liskov, “Practical Byzantine fault tolerance,” in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, pp. 173–186, New Orleans, NO, USA, February 1999.
- [10] C. Dwork, N. Lynch, and L. Stockmeyer, “Consensus in the presence of partial synchrony,” *Journal of the ACM*, vol. 35, no. 2, pp. 288–323, 1988.
- [11] L. Lamport, “Paxos made simple, fast, and byzantine,” in *Proceedings of the Sixth International Conference on Principles of Distributed Systems. OPODIS*, pp. 7–9, Reims, France, 2002.
- [12] F. Gai, F. Ali, J. Niu, C. Feng, F. Beschastnikh, and H. Duan, “Dissecting the performance of chained-BFT,” in *Proceedings of the D1CDCS*, pp. 595–606, Washington DC, USA, July 2021.
- [13] T. Hanke, M. Movahedi, and D. Williams, “Dfinity Technology Overview Series, Consensus system,” 2018, <https://arxiv.org/abs/1805.04548>.
- [14] T. H. H. Chan, R. Pass, and E. Shi, “PiLi: An Extremely Simple Synchronous Blockchain,” *Cryptology ePrint Archive*, vol. 2018, 2018.
- [15] I. Abraham, D. Malkhi, K. Nayak, L. Ren, and M. Yin, “Sync hotstuff: simple and practical synchronous state machine replication,” in *Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP)*, pp. 106–118, IEEE, San Francisco, CA, USA, May 2020.
- [16] A. Miller, Y. Xia, K. Croman, and E. Shi, “The Honey badger of BFT protocols,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 31–42, Vienna, Austria, October 2016.
- [17] I. Abraham, D. Malkhi, and A. Spiegelman, “Asymptotically Optimal Validated Asynchronous Byzantine agreement,” in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pp. 337–346, Toronto, Canada, July 2019.
- [18] A. Gągol, D. Leśniak, and D. Straszak, “Aleph: efficient atomic broadcast in asynchronous networks with byzantine nodes,” in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pp. 214–228, Świątek, Zurich, Switzerland, October 2019.
- [19] D. Malkhi, K. Nayak, and L. Ren, “Flexible Byzantine Fault tolerance,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1041–1053, London, UK, November 2019.
- [20] Z. Xiang, D. Malkhi, K. Nayak, and L. Ren, “Strengthened Fault Tolerance in Byzantine Fault Tolerant replication,” 2021, <https://arxiv.org/abs/2101.03715>.
- [21] J. Martin and L. Alvisi, “Fast byzantine consensus,” *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 3, pp. 202–215, 2006.
- [22] I. Abraham, K. Nayak, L. Ren, and Z. Xiang, “Brief Announcement: Byzantine Agreement, Broadcast and State Machine Replication with Optimal Good-Case Latency,” in *Proceedings of the 34th International Symposium on Distributed Computing (DISC 2020)*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, Research institute in Wadern, Germany, October 2020.
- [23] I. Abraham, K. Nayak, L. Ren, and Z. Xiang, “Good-case Latency of Byzantine Broadcast: A Complete Categorization,” 2021, <https://arxiv.org/abs/2102.07240>.
- [24] O. Naor, M. Baudet, D. Malkhi et al., “Cogsworth: Byzantine view synchronization,” 2019, <https://arxiv.org/abs/1909.05204>.
- [25] M. Baudet, A. Ching, A. Chursin et al., “State machine replication in the libra blockchain,” Technical Report D, 2019, <http://libra.axuer.com/docs/state-machine-replication-paper/>.
- [26] O. Naor and I. Keidar, “Expected Linear Round Synchronization: The Missing Link for Linear Byzantine SMR,” 2020, <https://arxiv.org/abs/2002.07539>.
- [27] V. Shoup, “Practical threshold signatures,” in *Proceedings of the Advances in Cryptology - EUROCRYPT 2000*, pp. 207–220, Zagreb, Croatia, May 2000.
- [28] A. Boldyreva, “Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme,” in *Proceedings of the International Workshop on Public Key Cryptography*, pp. 31–46, Springer, Miami, FL, USA, January 2003.
- [29] M. J. Fischer, N. A. Lynch, and M. Paterson, “Impossibility of distributed consensus with one faulty process,” *Journal of the ACM*, vol. 32, no. 2, pp. 1–7, 1983.

Research Article

ForkDec: Accurate Detection for Selfish Mining Attacks

Zhaojie Wang ¹, Qingzhe Lv ¹, Zhaobo Lu ¹, Yilei Wang ^{1,2} and Shengjie Yue ³

¹School of Computer Science, Qufu Normal University, Rizhao 276826, China

²Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin, 541004, China

³School of Information Science and Engineering, University of Jinan, Jinan 250022, China

Correspondence should be addressed to Yilei Wang; wang_yilei2019@qfnu.edu.cn

Received 25 October 2021; Accepted 19 November 2021; Published 30 November 2021

Academic Editor: Yuling Chen

Copyright © 2021 Zhaojie Wang 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.

Incentive mechanism is the key to the success of the Bitcoin system as a permissionless blockchain. It encourages participants to contribute their computing resources to ensure the correctness and consistency of user transaction records. Selfish mining attacks, however, prove that Bitcoin's incentive mechanism is not incentive-compatible, which is contrary to traditional views. Selfish mining attacks may cause the loss of mining power, especially those of honest participants, which brings great security challenges to the Bitcoin system. Although there are a series of studies against selfish mining behaviors, these works have certain limitations: either the existing protocol needs to be modified or the detection effect for attacks is not satisfactory. We propose the ForkDec, a high-accuracy system for selfish mining detection based on the fully connected neural network, for the purpose of effectively deterring selfish attackers. The neural network contains a total of 100 neurons (10 hidden layers and 10 neurons per layer), learned on a training set containing about 200,000 fork samples. The data set, used to train the model, is generated by a Bitcoin mining simulator that we preconstructed. We also applied ForkDec to the test set to evaluate the attack detection and achieved a detection accuracy of 99.03%. The evaluation experiment demonstrates that ForkDec has certain application value and excellent research prospects.

1. Introduction

Bitcoin is essentially a decentralized, distributed public ledger, which allows anyone or institution to participate in publishing transactions in a client-side manner [1]. The transaction will be collected by the participants (called miners) in the network and then added to the public ledger through a consensus protocol. The consensus protocol adopted by Bitcoin is called Proof-of-Work. All miners compete to solve a difficult-to-solve but easy-to-verify cryptographic puzzle. The miner who successfully solves the puzzle first is allowed to add transactions to the ledger and receive Bitcoin rewards [2]. Incentive mechanism is central to the functionality of Bitcoin, which ensures the security and liveness of Bitcoin by encouraging a large number of honest miners to participate in the consensus process [3]. Traditionally, it is believed that Bitcoin's incentive mechanism is incentive-compatible, but the emergence of selfish

mining proves that this opinion is inaccurate [2]. By strategically publishing previously withholding blocks to invalidate blocks mined by honest miners, selfish attackers can collect additional reward shares that should belong to honest miners. The harm of selfish mining attacks is not limited to this. Unfair reward distribution will induce some rational participants to be selfish. A large number of selfish participants may also launch collusive attacks to infringe the revenue of other honest participants, which will seriously damage Bitcoin's reputation. Resulting in plenty of honest miners quitting will weaken the security significantly and give other attacks (e.g., double-spending attacks) an opportunity to take advantage of. Although selfish mining attacks have not been discovered in the real world, with the continuous improvement of potential attackers' computing power and the iterative upgrade of attack algorithms [4–10], the possibility of this attack is gradually increasing. We consequently must attach great importance to the detection

of this attack to ensure that it can be discovered and countermeasures are taken as soon as possible when an attack occurs.

1.1. Related Works. Ethan Heilman proposed a method based on unforgeable timestamps against selfish mining [11], called *Freshness Preferred*. It requires miners to add unforgeable timestamps to blocks, and it invalidates the blocks withheld by attackers by encouraging honest miners to choose blocks with the latest timestamp. The disadvantage of this method, however, is that it requires a credible timestamp agency to generate unforgeable timestamps and requires honest miners to record all recent timestamp release logs. Solat et al. [12] proposed a new solution that does not use unforgeable timestamps, called the *ZeroBlock*. The idea is that if selfish miners withhold blocks for more than a preset time interval, all honest miners will directly reject the block. The *ZeroBlock* scheme forces the selfish attacker to be unable to withhold blocks for a long time. Zhang et al. proposed the *Weighted Fork-Resolving Policy*. When a fork occurs, a weight is calculated for each branch. And, it is recommended that honest miners no longer simply rely on the length of the branch when determining the main chain but choose the branch with the largest weight [13]. Saad et al. [14] assigned an expected confirmation height (i.e., the expected height of the block containing the specified transaction) to each transaction by measuring the transaction size, transaction fee, and other factors. The smaller the gap between the actual confirmation height and the expected height, the lower the possibility of selfish mining behavior. Lee et al. increased the profit threshold of selfish mining from 25% to 33% by adding transaction creation time to the transaction data structure [15]. Chicarino et al. [16] analyzed the impact of selfish mining on Bitcoin’s fork height and judged whether a selfish mining attack occurred by monitoring the abnormal changes in the fork height.

1.2. Motivation and Contribution. Since Eyal and Sirer proposed the concept of selfish mining and pointed out its harmfulness; a series of studies on this attack have appeared [4, 10, 17, 18, 19, 20]. The main focus of most research, however, is to increase the attacker’s rewards or reduce the mining power threshold. By contrast, there are relatively few research studies on selfish mining defense measures [11–16], and many works require upgrading the existing protocol, which is costly to implement. The selfish mining detector [16] proposed by Chicarino et al. realized the detection of selfish mining without modifying the Bitcoin protocol. However, it only considers the factor of fork height and does not take other factors into consideration, which leads to a certain misjudgment rate. To improve the detection accuracy, in this work, we propose a selfish mining attack detection system based on a machine learning classification model, called ForkDec. The system can detect selfish mining attacks in the Bitcoin network with an accuracy rate of 99.03%. Our primary contributions are threefold as follows:

- (1) We construct a data set containing approximately 200,000 fork samples. Considering that selfish mining has not been discovered in reality, we build a Bitcoin mining simulator to simulate the Bitcoin mining process in the presence of propagation delays and selfish miners. In the simulation process, the simulator records all the fork features, and then the feature extractor extracts feature vectors based on the fork features to construct fork samples.
- (2) We present ForkDec as an accurate detection system for detecting selfish mining attacks in Bitcoin. To accurately detect selfish mining, we trained a classification model based on logistic regression and a fully connected neural network (with 10 hidden layers and 10 neurons per layer) on the training set, respectively, and then applied the learned model to ForkDec for attack detection.
- (3) We applied ForkDec to the test set to evaluate its performance. The evaluation results show that ForkDec is better than the selfish mining detector [16] in accuracy. In addition, we also found that the classification model based on the fully connected neural network has a better overall performance.

1.3. Roadmap of This Paper. The rest of the paper is organized as follows. In Section 2, we introduce the details of the ForkDec system, including the construction of the data set and the selection of the classification model. In Section 3, the evaluation results and discussion of the proposed system are given. Finally, we conclude this work in Section 4.

2. ForkDec: System Description

Figure 1 presents the basic architecture of the ForkDec system. It mainly includes three modules: data set construction, model training, and attack detection. Firstly, we built a simulator to simulate the Bitcoin network with selfish attackers. The simulator will record the information of each block (block height, miner, and timestamp), especially fork features, and then each fork will be delivered to the feature extractor to extract the feature vector to construct the fork data set. We, subsequently, use the built training set to train the classification model and embed the learned model into ForkDec for attack detection.

2.1. Data Set Construction. The classification model relies on the training set to learn sample features and to identify unknown samples. To get an excellent attack detection model, we must have a training set with abundant selfish mining samples. Since machine learning has not been applied to selfish mining detection before, there is no existing data set available. To solve this issue, we constructed a data set containing 200,000 fork samples for model training, in which the ratio of natural fork samples to attacking fork samples is 3 : 7.

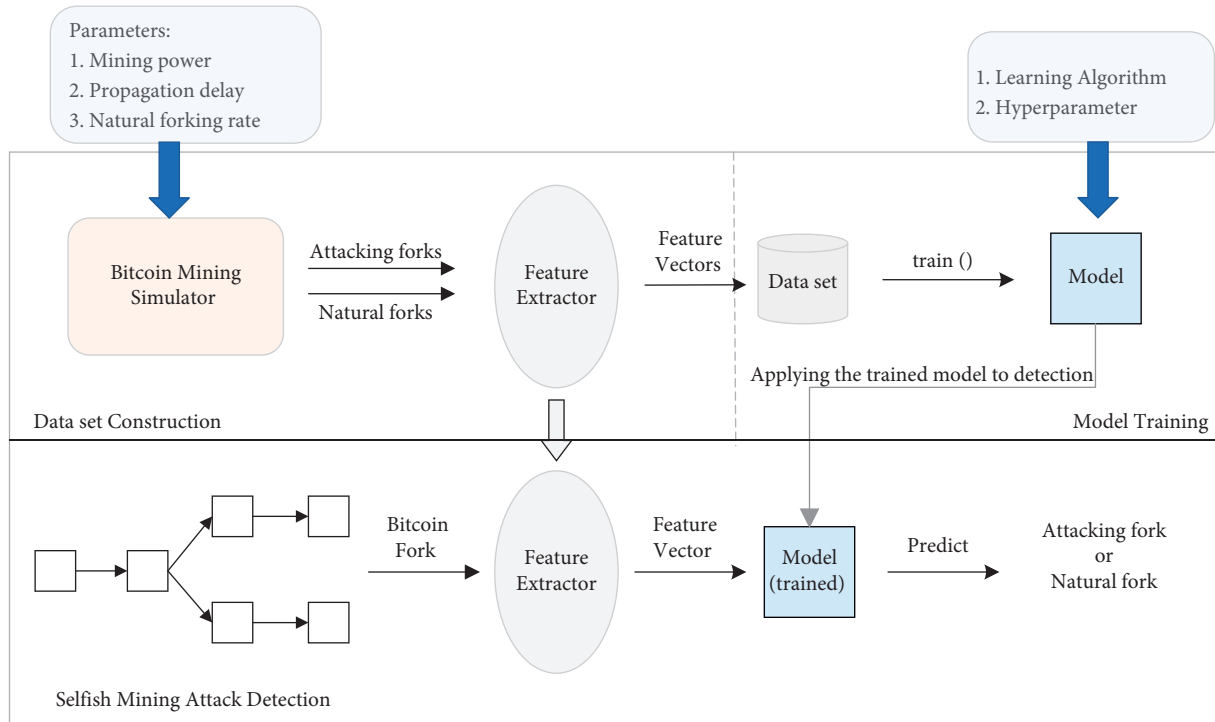


FIGURE 1: Schematic of the ForkDec detection system.

2.1.1. Feature Vector Extraction. All miners in the Bitcoin network utilize Proof-of-Work to compete for accounting rights to create new blocks at an average rate of 10 minutes. After being created, the new block will be broadcast immediately by all honest miners in the Peer-to-Peer network. Unlike honest miners, the selfish attacker will secretly withhold newly mined blocks to create conflicting branches. Then, the attacker invalidates the blocks mined by honest counterparts through strategically publishing the withheld blocks. In this way, the attacker could increase his proportion of rewards distribution. By studying the strategy of the selfish attacker, it can be known that the attacker carries out attacks by making forks. Therefore, the key to detecting this type of attack is to track the fork data in the blockchain. Based on this, we construct a feature extractor to represent the fork data as a feature vector. The classification model learns the characteristics of the selfish mining attack through the feature vector, thereby detecting the attacks. In the Bitcoin, we define the structure of the feature vector as follows: $\{h, l, i_b, i_t\}$. The meaning of each feature is as follows:

- (i) h is the block height of the fork
- (ii) l is the length of the fork, i.e., the number of blocks on the conflicting branch
- (iii) i_b is the number of blocks between this fork and the previous fork
- (iv) i_t is the absolute value of the difference between the timestamps of the first block of each branch

Subsequently, we use an example to present the general process of feature vector extraction, as illustrated in Figure 2. For simplicity and without loss of generality, we

assume that b_0 is the first block after the previous fork is resolved, and its timestamp is t_0 . After b_1 is accepted by all participants, two valid blocks b_2 (with timestamp t_2) and b'_2 (with timestamp t'_2) are propagated in the P2P network. Consequently, the blockchain makes a fork since b_2 and b'_2 have the same block height, i.e., $h(b_2)$. Note that we utilize $h(x)$ to indicate the height of block x . The Bitcoin mining simulator will capture and record information about this fork. Then, the extractor converts this information into a 4-dimensional vector, which is the feature vector on the far-right side of Figure 2.

2.1.2. Fork Sample Generation. Under the setting that only considers selfish mining attacks, there are two types of forks in the Bitcoin network: natural forks and attacking forks. Natural fork means that when a block is propagated in the network, other miners create and broadcast a block with the same height, which leads to inconsistencies in the distributed ledger. This inconsistency is not caused by the attack but by network propagation delays [21]. Christian Decker and Roger Wattenhofer pointed out that the average delay of a block in Bitcoin is 12.6 seconds, and after the new block is broadcast for 40 seconds, 95% of the nodes have received the block [21]. In other words, the timestamp difference between most conflicting blocks in the Bitcoin network is close to the average propagation delay. Based on this, we adopt an exponential distribution with the expected value of 12.6 seconds to approximate the block propagation delay distribution, as shown in Figure 3. The simulator then randomly samples based on the distribution to simulate the timestamp interval of a natural fork.

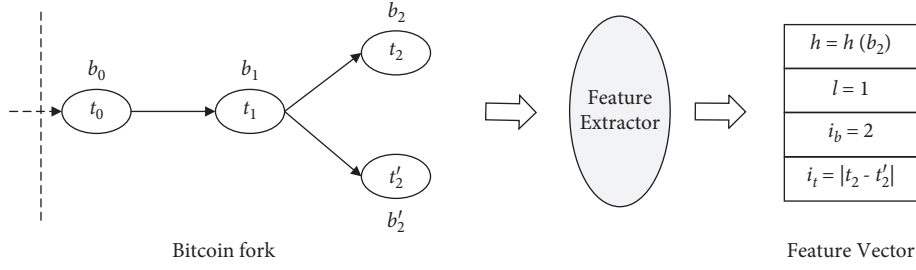


FIGURE 2: A clearly expressed example of feature vector extraction.

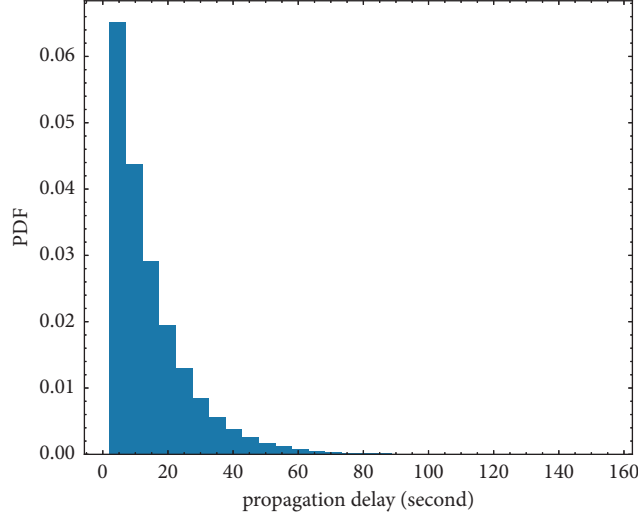


FIGURE 3: The sampling distribution of propagation delay.

The opposite is the attacking fork which is caused by a malicious attack. Figure 4 shows the formation of an attacking fork. Assume that the selfish attacker firstly mines block A_1 at time t . According to the SM1 strategy [2], the attacker will secretly withhold block A_1 . Since honest miners will not perceive the existence of A_1 until it is published, the honest miners may mine a new block A_2 at any time t' after time t . Then, there is $0 < t' - t$; considering the average block creation time is 600 seconds (10 minutes), we set $0 < t' - t \leq 600$. That is when the simulator is simulating an attacking fork, the timestamp interval of conflicting blocks is randomly sampled between 0 and 600 seconds.

2.2. Classification Model. The selection of the learning algorithm is another key point for ForkDec to realize high-accurate detection. It is impossible to get an efficient model if the learning algorithm is not well selected and even if there are rich sample data sets to utilize. We, respectively, test the detection effect of ForkDec when logistic regression and a fully connected neural network are used as the classification model. Among them, the logistic regression features a faster model convergence rate while the fully connected neural network performs better in accuracy rate.

2.2.1. Logistic Regression. Logistic regression is a classification model that utilizes a linear model to predict binary classification problems. The idea is to map the output of the

linear model (any continuous value) to a value between 0 and 1 by adding the sigmoid function after the linear model. Equation (1) presents the mathematical expression of logistic regression, where x^T represents the sample to be classified, (w, b) represents the model parameter, and \hat{y} represents the prediction results of the model (also called the confidence level):

$$\begin{aligned} \hat{y} &= \text{Sigmoid}(x^T w + b) \\ &= \frac{1}{1 + \exp(x^T w + b)}. \end{aligned} \quad (1)$$

By setting the threshold to 0.5, the ForkDec classifies fork samples with a confidence level of more than 0.5 as attacking forks, otherwise as natural forks. In addition, to prevent overfitting, we use minimizing the cost function (with the L2 penalty term) as the optimization problem during model training and then apply the L-BFGS algorithm, a kind of quasi-Newton method, to solve the optimization problem.

2.2.2. Fully Connected Neural Network. Logistic regression has the characteristics of clear structure and simplicity. However, on the other hand, a simple model may not be able to make full use of the rich training samples and cannot achieve top-notch detection results. To further improve the accuracy in attack detection, we additionally consider the use

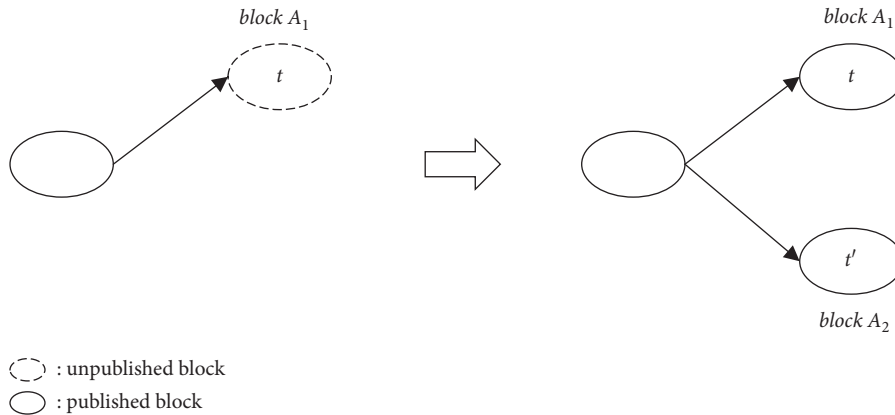


FIGURE 4: The example of an attacking fork.

of fully connected neural network, also known as multilayer perceptron, as the classification model. Figure 5 presents the structure of the fully connected neural network.

The input layer on the far left is composed of a group of neurons $\{x_i | x_1, x_2, \dots, x_m\}$ representing the characteristics of the sample. Unlike logistic regression, there can be one or more nonlinear layers between the input layer and output layer of a neural network, called hidden layers. The neurons in each hidden layer perform a weighted linear summation conversion on the values of the previous layer. The converted value firstly passes through the activation function and then is delivered to the next layer until the final output layer. In the ForkDec system, we utilize backpropagation to train the neural network, and finally, we get a fully connected neural network with 10 hidden layers and 10 neurons in each layer.

3. Evaluation

In this section, we evaluate the performance of ForkDec in detecting selfish mining attacks. The ForkDec system utilizes Scikit-learn (version 1.0) to implement the model training. Scikit-learn, an open-source and efficient machine learning tool library, is implemented based on the Python program language. Subsequently, we embed the trained model into the ForkDec system and test it on a test set containing 76,686 samples. The test results show that the ForkDec system can achieve a detection accuracy of 99.03% when the fully connected neural network is used as the classification model and 98.76% when using logistic regression. We additionally compare the performance of the ForkDec detection system with the selfish mining detector (hereinafter referred to as SM detector) proposed in [16]. We also train the fully connected neural networks under different hyperparameters to find the optimal model and then detect selfish attackers with different abilities.

3.1. Comprehensive Performance. By applying ForkDec to a test set containing 76,686 samples, the confusion matrix of ForkDec in detecting selfish mining attacks can be obtained, which is presented in Table 1. In the confusion matrix, the classification results of ForkDec and the real distribution of the samples are shown, where positive represents the attacking fork category and negative represents the natural

fork category. To facilitate the description, we name the ForkDec system, respectively, according to the different classification models:

- (i) ForkDec-DNN is the ForkDec system with the fully connected neural network as the classification model
- (ii) ForkDec-LR is the ForkDec system with logistic regression as the classification model
- (iii) ForkDec is the collective name of ForkDec-DNN and ForkDec-LR

From Table 1, we can see that the advantage of ForkDec-DNN is that it does not misclassify natural forks as attacking forks, while ForkDec-LR misclassifies 542 natural forks as attacking forks. However, ForkDec-DNN also has its disadvantages; that is, 745 attacking forks are misidentified as natural forks by ForkDec-DNN, while this value is only 407 for ForkDec-LR.

To more intuitively evaluate the performance of ForkDec, we present the accuracy, precision, recall, and F_1 value of ForkDec on the test set in Figure 6. The meanings of these indicators are as follows:

- (i) Accuracy: it is the proportion of correctly classified samples to the total sample.
- (ii) Precision: among all attacking fork samples detected by the model, precision is the proportion of real attacking samples.
- (iii) Recall: among all the attacking samples, recall is the proportion detected by the model.
- (iv) F_1 : the F_1 value, shown in (2), can be used to measure the comprehensive performance of the model in terms of precision and recall. The reason is that the F_1 value is only high when both precision and recall are high:

$$\frac{1}{\text{precision}} + \frac{1}{\text{recall}} = \frac{2}{F_1}. \quad (2)$$

It can be found in Figure 6 that ForkDec-DNN and SM Detector both have top scores in precision rate, which

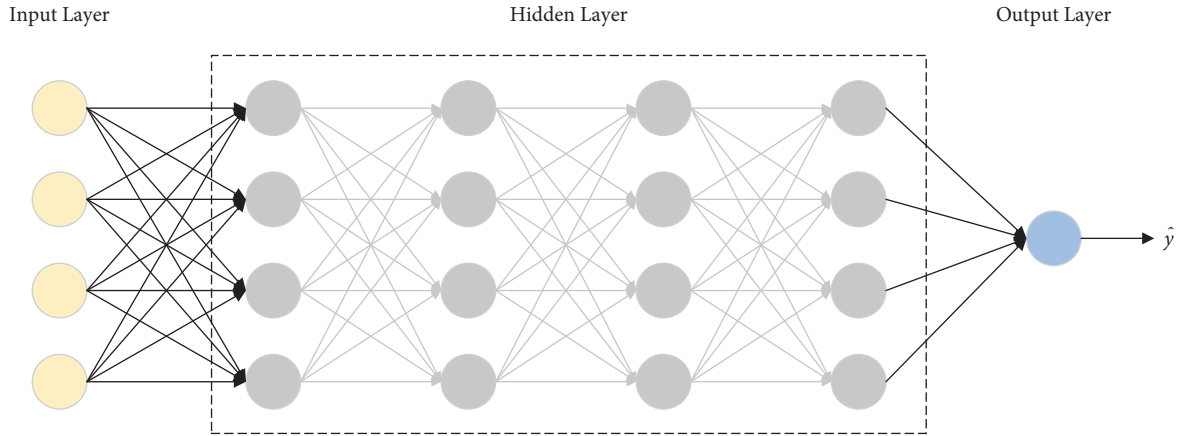


FIGURE 5: The example of a fully connected neural network with 4 hidden layers and 4 neurons per layer.

TABLE 1: The confusion matrix of ForkDec. In the values (x, y) , x indicates the classification result of ForkDec-DNN and y represents the result of ForkDec-LR.

The real distribution of samples	The classification results of ForkDec	
	Positive	Negative
Positive	(57238, 57576)	(745, 407)
Negative	(0, 542)	(18703, 18161)

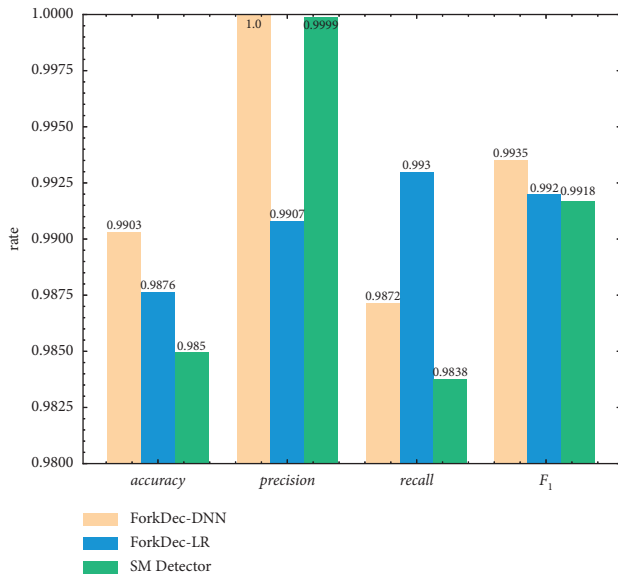


FIGURE 6: The performance of ForkDec-DNN, ForkDec-LR, and SM Detector.

indicates that both can ensure that there are almost no false positives in all detected attacking forks. Precision rate and recall rate are a pair of contradictory indicators. ForkDec-DNN and SM Detector pursue the ultimate precision rate, which also means that both will have a loss in the recall rate. However, the loss of SM Detector's recall rate is greater, so this leads to a lower F_1 value of SM Detector. Moreover, ForkDec-DNN also has the highest accuracy rate among the three, which cannot be achieved by SM Detector. Unlike ForkDec-DNN's extreme performance in precision rate,

ForkDec-LR balances various indicators. In particular, ForkDec-LR has the highest recall rate among the three. In other words, ForkDec-LR can detect attacking forks as many as possible, with only a few false negatives.

3.2. The Optimal Model. To find the optimal model, we train the fully connected neural networks under different hyperparameters. Then, we apply these trained models to the test set. The performance of these models on the test set is shown in Table 2. It can be concluded from Table 2 that a neural network with 10 hidden layers and 10 neurons per layer has the best performance. And, more neurons do not mean better classification performance. It is worth mentioning that a neural network with 10 hidden layers and 10 neurons per layer may not be optimal, but its performance is close to the optimal model.

3.3. Detection for Attacker with Varying Power. In order to fully evaluate the detection effect of ForkDec, we additionally considered the detection of selfish attackers under specific mining power. We first utilize α to represent the fraction of attacker's mining power in the power of the entire Bitcoin network. Figure 7 presents the detection effect of ForkDec against different power attackers. We notice that the accuracy rate, recall rate, and F_1 value drop rapidly when $\alpha > 0.25$. This is because, as the attacker's mining power increases, the frequency of the selfish mining attack is getting higher and higher, resulting in a large number of forks with close timestamps in the blockchain. Many of these forks are not correctly detected by the model, leading to a drop in recall rate, as the characteristics of such attacking forks are very similar to natural forks. Then, the accuracy rate and F_1

TABLE 2: The performance of neural networks with different hyperparameters on the test set.

Model	Accuracy	Precision	Recall	F_1
8 layers \times 10 neurons	0.9902329	1.0	0.9870824	0.9934992
9 layers \times 10 neurons	0.9902459	1.0	0.9870997	0.9935080
10 layers \times 10 neurons	0.9902851	1.0	0.9871514	0.9935342
11 layers \times 10 neurons	0.9902590	1.0	0.9871169	0.9935167
12 layers \times 10 neurons	0.9902459	1.0	0.9870997	0.9935080
12 layers \times 12 neurons	0.9902199	1.0	0.9870652	0.9934905

m layers \times n neurons indicates a neural network with m hidden layers and n neurons per layer.

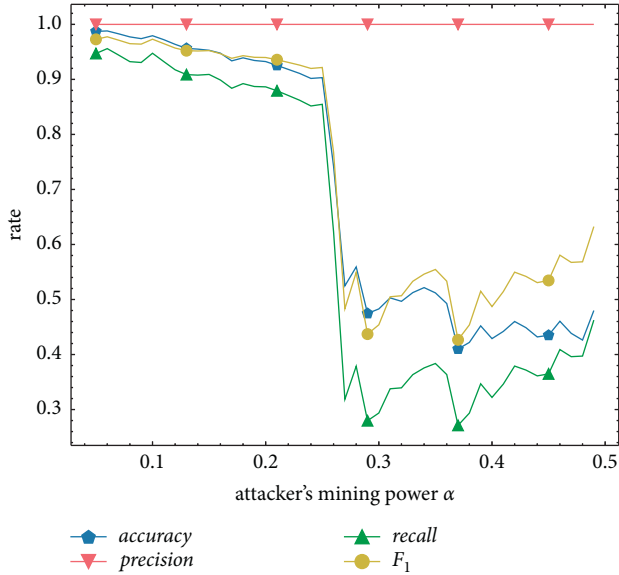


FIGURE 7: The detection of selfish attacker with varying mining power.

value also drop. However, even in the face of powerful attackers, ForkDec still maintains a very high-accuracy rate. It can still ensure that there are almost no false positives during the detection process.

4. Conclusion

In this work, we propose a detection system for selfish mining attacks in Bitcoin, called ForkDec. The system is based on the machine learning classification model to realize intelligent detection of attacks. To ensure that ForkDec has a high detection accuracy, we construct a data set containing about 200,000 Bitcoin fork samples for model training. We then apply ForkDec to the test set for evaluation. The evaluation results show that ForkDec can achieve an accuracy of 99.03% for the detection of selfish mining in Bitcoin. What needs to be clear is that ForkDec can only detect the presence of an attack but cannot identify the miner who launched the attack. In future work, we will further analyze the attacker's strategy and improve ForkDec to accurately locate the attacker. In addition, the blockchain also applies in the fields of privacy protection [22] and data traceability. Attackers may use other methods to attack the blockchain. Hence, we also have to study the application of ForkDec to the detection of other attacks, e.g., double-

spending attacks [23], time-bandit attacks [24], and blockchain DoS attacks [25].

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 there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the Foundation of National Natural Science Foundation of China (Grant nos. 62072273, 72111530206, 61962009, 61873117, 61832012, 61771231, and 61771289); Natural Science Foundation of Shandong Province (Grant no. ZR2019MF062); Shandong University Science and Technology Program Project (Grant no. J18A326); Guangxi Key Laboratory of Cryptography and Information Security (Grant no. GCIS202112); Major Basic Research Project of Natural Science Foundation of Shandong Province of China (Grant no. ZR2018ZC0438); Major Scientific and Technological Special Project of Guizhou Province (Grant no. 20183001); Foundation of Guizhou Provincial Key Laboratory of Public Big Data (Grant no. 2019BD-KFJJ009); and Talent Project of Guizhou Big Data Academy, Guizhou Provincial Key Laboratory of Public Big Data (Grant no. [2018]01).

References

- [1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Decentralized Business Review*, Article ID 21260, 2008.
- [2] I. Eyal and E. G. Sirer, "Majority is not enough: bitcoin mining is vulnerable," in *Proceedings of the International conference on Financial Cryptography and Data Security*, pp. 436–454, Kota Kinabalu, Malaysia, February 2014.
- [3] C. Hou, M. Zhou, Y. Ji et al., "SquirRL: automating attack analysis on blockchain incentive mechanisms with deep reinforcement learning," 2019, <https://arxiv.org/abs/1912.01798>.
- [4] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Proceedings of the International conference on Financial Cryptography and Data Security*, pp. 515–532, Christ Church, Barbados, February 2016.

- [5] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "Be selfish and avoid dilemmas: fork after withholding (faw) attacks on bitcoin," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 195–209, Dallas TX USA, October 2017.
- [6] S. Gao, Z. Li, Z. Peng, and B. Xiao, "Power adjusting and bribery racing: novel mining attacks in the bitcoin system," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 833–850, Auckland, New Zealand, July 2019.
- [7] L. Tao, W. Zhaojie, Y. Guoyu, and C. Yang, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, pp. 3596–3612, 2021.
- [8] L. Tao, C. Yuling, W. Yanli et al., "Rational protocols and attacks in blockchain system," *Security and Communication Networks*, vol. 2020, Article ID 8839047, 11 pages, 2020.
- [9] G. Yang, Y. Wang, Z. Wang, Y. Tian, X. Yu, and S. Li, "IPBSM: an optimal bribery selfish mining in the presence of intelligent and pure attackers," *International Journal of Intelligent Systems*, vol. 35, no. 11, pp. 1735–1748, 2020.
- [10] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: generalizing selfish mining and combining with an eclipse attack," in *Proceedings of 2016 IEEE European Symposium on Security and Privacy*, pp. 305–320, Saarbruecken, Germany, March 2016.
- [11] E. Heilman, "One weird trick to stop selfish miners: fresh bitcoins, a solution for the honest miner (poster abstract)," in *Proceedings of the International conference on Financial Cryptography and Data Security*, pp. 161–162, Okinawa, Japan, April 2014.
- [12] S. Solat and M. Potop-Butucaru, "Zeroblock: timestamp-free prevention of block-withholding attack in bitcoin," 2016, <https://arxiv.org/abs/1605.02435>.
- [13] R. Zhang and B. Preneel, "Publish or perish: a backward-compatible defense against selfish mining in bitcoin," in *Proceedings of Cryptographers' Track at the RSA Conference*, pp. 277–292, San Francisco, CA, USA, February 2017.
- [14] M. Saad, L. Njilla, C. Kamhoua, and A. Mohaisen, "Countering selfish mining in blockchains," in *Proceedings of 2019 International Conference on Computing, Networking and Communications*, pp. 360–364, Honolulu, HI, USA, February 2019.
- [15] J. Lee and Y. Kim, "Preventing bitcoin selfish mining using transaction creation time," in *Proceedings of 2018 International Conference on Software Security and Assurance*, pp. 19–24, Seoul, Korea, July 2018.
- [16] V. Chicarino, C. Albuquerque, E. Jesus, and A. Rocha, "On the detection of selfish mining and stalker attacks in blockchain networks," *Annals of Telecommunications*, vol. 75, no. 3–4, pp. 143–152, 2020.
- [17] L. Tao, W. Zhaojie, C. Yuling, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, .
- [18] K. A. Negy, P. R. Rizun, and E. G. Sirer, "Selfish mining re-examined," in *Proceedings of the International conference on financial cryptography and data security*, Kota Kinabalu, Malaysia, February 2020.
- [19] G. Cyril and R. Pérez-Marco, "On profitability of selfish mining," 2018, <https://arxiv.org/abs/1805.08281>.
- [20] R. B. Zur, I. Eyal, and A. Tamar, "Efficient MDP analysis for selfish-mining in blockchains," in *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pp. 113–131, New York, NY, USA, October 2020.
- [21] C. Decker and R. Wattenhofer, "Information propagation in the Bitcoin network," in *Proceedings of IEEE P2P*, pp. 1–10, Trento, Italy, September 2013.
- [22] C. Yuling, S. Jing, Y. Yixian, T. Li, X. Niu, and H. Zhou, "PRSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, .
- [23] M. Rosenfeld, "Analysis of hashrate-based double spending," 2014, <https://arxiv.org/abs/1402.2009>.
- [24] P. Daian, S. Goldfeder, T. Kell et al., "Flash boys 2.0: front-running, transaction reordering, and consensus instability in decentralized exchanges," 2019, <https://arxiv.org/abs/1904.05234>.
- [25] M. Mirkin, Y. Ji, J. Pang, A. Mundt, I. Eyal, and A. Juels, "BDoS: blockchain denial-of-service," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 601–619, New York, NY, USA, November 2020.

Research Article

A Research on Traceability Technology of Agricultural Products Supply Chain Based on Blockchain and IPFS

Lejun Zhang ^{1,2}, Weimin Zeng,¹ Zilong Jin,³ Yansen Su,⁴ and Huiling Chen⁵

¹College of Information Engineering, Yangzhou University, Yangzhou 225127, China

²Cyberspace Institute Advanced Technology, Guangzhou University, Guangzhou 510006, China

³School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 21004, China

⁴Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education,

School of Computer Science and Technology, Anhui University, Hefei 230601, China

⁵Department of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China

Correspondence should be addressed to Lejun Zhang; zhanglejun@yzu.edu.cn

Received 10 August 2021; Revised 22 October 2021; Accepted 27 October 2021; Published 12 November 2021

Academic Editor: Xiu-Bo Chen

Copyright © 2021 Lejun Zhang 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 technology, the fundamental technology of Bitcoin, is featured with high transparency, decentralization, traceability, tamperproof nature, and anonymousness. In this thesis, a case study of the traceability of agricultural products is to explain a traceability solution of agricultural products supply chain based on blockchain and IPFS. The latter one is used to store large quantities of transactions data; and the former one is used for the safety of data storage and circulation. And consumers can know the quality of agricultural products in the shortest time through the evaluation function. As shown in the experiment, the solution is more efficient and secure compared with existing supply chain traceability methods, meeting the traceability requirements of security, transparency, and reliability. Furthermore, the traceability, safety, and performance of the scheme are also analyzed here.

1. Introduction

The consumption market in China is facing a brand new era of traceability, and product traceability has become a hot issue concerned by the society. Food security has become an escalating concern in the society. Even though national traceability standards for major products have been formulated by the government, incidents of counterfeit and inferior products often occur in the market. Thus, a series of food security problems have triggered the consumer trust crisis, which is also a major challenge to the efforts made in the progress of the national development of a credible society [1]. The food traceability system can identify the source of food and detail the whole process from food production to dining tables. In case of any food security and quality problem, it can quickly locate the key link of the problem and identify the subject of responsibility to contain the problem from worsening,

which provides an effective way to solve the food security problems.

The analysis of the current food information storage platforms and supply chains reveals that the current food traceability system is deficient with the following defects: It highly relies on centralized databases, exposing a hidden hazard of information tampering to many key links, such as storage, presentation, and maintenance of data[2]; The phenomenon of “information silo” undermines the current supply chains, as the internal systems of the entities possess most of the information. While the lateral interconnection between the systems is insufficient which makes it difficult to realize the linkage regulation, in the whole circulation process from food production to consumer consumption, the extent and efficiency of automation are insufficiently low in the links of food processing, warehousing, logistics, etc. With broad application of 5G technology, the demand for data storage is surging sharply, and the market is facing

overwhelming pressure with larger scales of data storage requirements. The emergence of IPFS is right on time. The *Notice of the General Office of the National Radio and Television Administration on Issuing: A Series of White Papers on the Application of Blockchain Technology* has repeatedly mentioned the distributed storage of IPFS and blockchain, affirming the application value and technical advantages of IPFS. The blockchain technology can magnify the function of IPFS, while IPFS can overcome the data storage constraints of blockchain. The combination of the two technologies is the trend of food traceability development in the future. In this thesis, grain traceability, as an example, is introduced to demonstrate the real-time monitoring of the supply chain, in which case effective tracing of grains and traceability of business transactions of agricultural products in the supply chain are realized by virtue of smart contracts deployed in the blockchain, in a bid to carry out real-time monitoring of the supply chain and improve transparency, and we call it Bc-IPFS.

The main contributions can be summarized as follows:

- (i) Combine blockchain technology and IPFS technology. The details of transactions are stored in IPFS and the hash is stored in the blockchain, which cannot only ensure data security and effectively overcome the shortcoming of constrained data storage capacity of blockchain.
- (ii) Discuss the sequence of interaction and relations between major participants, the solution, and the key points resolved.
- (iii) Propose the transaction assessment function and make the consumer information and purchased products private.
- (iv) Deploy smart contracts based on Hyperledger to realize traceability of the food supply chain and verify the feasibility by the throughput capacity test and delay test.

2. Relevant Work

With the development of the blockchain technology, the unique decentralization, traceability, and tamperproof nature of blockchain have been promoting the transition from traditional traceability to blockchain traceability [3]. More and more scholars start to study blockchain-based food supply chain traceability. Yu and Huang [4] put forth the traceability solution for broiler chickens by combining the blockchain technology and RFID technology. With the solution, smart devices can be used to scan the traceable QR code on the chicken claw ring to retrieve the corresponding data and information, where the chicken claw ring is designed into an “inverted tooth” shape to prevent its secondary use. Tian et al. [5] developed an agricultural food supply chain traceability system, covering the whole process of data acquisition and information management of all links of the entire supply chain. The RFID technology is adopted to realize data acquisition, data circulation, and data sharing, and the blockchain technology is adopted to ensure data

reliability. However, the RFID technology is deficient in high costs, such as the equipment costs of RFID transmitters, readers, and antennas. Besides, the availability of RFID frequency bands is varied in different countries. RFID is prone to inciting privacy leakage and other problems, and RFID can be easily impacted in an environment containing metal and moisture; thus RFID cannot be broadly utilized in large scale. Afterwards, Tian [6] proposed the food supply chain traceability based on hazard analysis and key control points (HACCP) by adoption of blockchain and Internet of things. Highly similar to the application scenario of [5], it adopts RFID for data acquisition, blockchain technology for ensuring data security, HACCP for monitoring and tracing of supply chains, and BigchainDB for storage and management of food supply chains data. However, on the one hand, BigchainDB is still exposed to the deficiency of RFID, and, on the other hand, it is not ideal for file storage, but for the structural data. Yang et al. [7] used Hyperledger as the traceability chain to store information in the local database, which is useful in solving the problem of blockchain deficiency in massive data storage. However, it is disadvantageous in high cost, slow data transition rate, low security, etc., in comparison with data storage by IPFS. Further, it does not provide the consumer with feedback function, so retailers cannot get access to product security and other aspects in the first time. Xie et al. [8] utilized the IoT technology to carry out ETH-based tracing of agricultural products, ensuring that data will not be maliciously tampered or damaged. However, on the data storage layer, data storage is blockchain-based; thus the network overheads will become increasingly greater with the increase of data volume. Hao et al. [9] researched the traceability storage solution based on the blockchain technology, which stores the crop growth information in IPFS and provides analysis of crop growth data by virtue of the auxiliary database. Although the solution overcomes the data storage constraint of blockchain, the focus of the system is on the acquisition of crop growth information, and thus the solution is not favorable to the information tracing subsequent to crop processing. Besides, the traceability of agricultural product supply chains includes the crop growth information and also the data and information subsequent to crop processing; thus traceability becomes a zero-distance shortcut from farmlands to dining tables. Salah et al. [10] researched the business transaction implementation method relying on ETH-based smart contract, in order to realize the traceability and transparency of soybean supply chain. However, due to the lack of consumer feedback function, retailers cannot gain access to safety problems in the first time after food is sold to consumers, and ETH involves data exploiting processes which consume time and resources.

With the development of blockchain 2.0, smart contract has been widely applied, and this thesis discusses the realization of traceability function automation and the introduction of the consumer feedback function based on the smart contract deployed on the blockchain. It is intended that, in case of any agricultural product security problem, entities in the supply chains may respond in the first time.

3. A Agricultural Product Supply Chain Traceability Solution Based on Bc-IPFS

In terms of the traceability of agricultural product supply chains, there are high requirements for the backup of transaction data. The IPFS storage technology is to separate a file into many pieces scattered on different locations of the network, which provides more powerful backup capacity compared with cloud storage. The blockchain technology can ensure the integrity of the data stored in IPFS, which is ideal for the traceability of agricultural product supply chains. Thus, in this section, we use the Hyperledger to trace and implement the transactions in the agricultural product supply chains by deployment of smart contracts of chaincode and store transaction information in IPFS to effectively reduce the reliance on the centralized database. Store the hash in the blockchain to take advantage of the features of blockchain to provide secure and reliable transaction records for the management of supply chains and thus ensure the information authenticity and reliability of the agricultural product acquired by the consumers.

3.1. Schematic Design. The unique feature of the solution is that it adopts the blockchain as the foundation layer, allows transactions between mutually untrusted users through smart contract, and adopts the IPFS technology to resolve the data storage constraint of blockchain. Within a certain period after the closure of a transaction, the consumer may use the ring signature algorithm to carry out anonymous assessment on the retailer. Relevant regulatory authorities or product suppliers may determine the quality security problems of a certain batch of products in the first time through consumer feedback. The batch information of the product purchased by the consumer can be used to identify the specific product batch, and the smart contract can be used to trace the root cause.

In this thesis, we propose the deployment of smart contract on Hyperledger, which can automatically send the preset data resources (including the triggering condition event) according to the contract information agreed in the smart contract when the triggering condition is met. Once the smart contract is deployed, it cannot be changed but can be upgraded to launch new functions or fix bugs [11]. The Fabric smart contract is independent of the underlying ledger, and it is not required to relocate the ledger data to the new smart contract when the smart contract is upgraded, which truly realizes the separation between logic and data. The smart contract of Fabric is referred to as chaincode, including system chaincode and user chaincode [12]. System chaincode is used to realize system-level functions and the processing logics of Fabric nodes, including system configuration, endorsement, and verification [13, 14]. User chaincode operates in an isolated chaincode container and is responsible for the user's application function, providing status processing logics based on the distributed blockchain ledger. It is programmed by application developers as a support to upper-level services. Smart contract receives transactions and triggers events in the form of function call,

so that a participating entity can constantly monitor the events being sent in blockchain without too many expenses [15].

Figure 1 shows the traceability of agricultural products. The Hyperledger smart contract is adopted to record information, and all participants involved in the supply chain are added to the processes, which are used to trace the agricultural products from the place of origin to the end consumers in a digital manner. The supervisory and regulatory bodies or relevant government departments deploy the main smart contract which provides an interface to the entities in the supply chain for function call and transaction implementation. IPFS hash files are stored in blockchain, which can be processed within the specified time period if the product conforms to the buyer's requirements. Supervisory and regulatory bodies almost deal with the entire supply chain. For instance, the agricultural bureau may carry out supervision, recording and management of farmer information, seed information, product information, etc., to ensure information authenticity. The quality supervision bureau must carry out supervision, management, and recording of processing plant information, retailer information, and product quality information and ensure security and quality of agricultural products. In order to improve the storage capacity, the information of all products and the data of all transactions and events are stored in IPFS, and the blockchain is only used to store the hash value of the data. IPFS is designed for distributed storage, which can be combined with Hyperledger to improve its throughput. The formula symbol description is shown in Table 1. Each transaction (TX_{tr}) bears the product identifier (ID_{pro}), product data hash (H_{pro}), identifier (ID_{own}) of the product owner and its signature (Sig_{own}), and public key (PK_{own})

$$\begin{cases} TX_{tr} = [ID_{pro}||H_{pro}||ID_{own}||Sig_{own}||PK_{own}], \\ H_{pro} = [P_{typ}||P_{quan}||P_{pri}||P_{ori}]. \end{cases} \quad (1)$$

TX_{tr} is stored in IPFS and hash is stored in blockchain. The product hash (H_{pro}) includes the product type (P_{typ}), quantity (P_{quan}), price (P_{pri}), and place of origin (P_{ori}). When a product is confirmed to have been delivered from the seller to the buyer in a transaction, $TX_{tr} = [ID_{pro}||H_{pro}||ID_{buy}||Sig_{buy}||PK_{buy}||Sig_{sell}||PK_{sell}]$, where ID_{buy} , Sig_{buy} , and PK_{buy} represent the identifier, signature, and public key of the owner, respectively;

$$TX_{tr} = [ID_{pro}||H_{pro}||ID_{buy}||Sig_{buy}||PK_{buy}||Sig_{sell}||PK_{sell}]. \quad (2)$$

ID_{buy} , Sig_{sell} , and PK_{sell} represent the identifier, signature, and public key of the seller, that is, the signature (Sig_{own}) and public key (PK_{own}) of the product owner in (1). The identity is required to be transformed in the process of product transaction, the owner of a transaction will be the seller in a subsequent transaction, and the buyer will become the owner of the product when the transaction is complete. After the consumer buys the product from the retailer, the seller creates the transaction order m_R

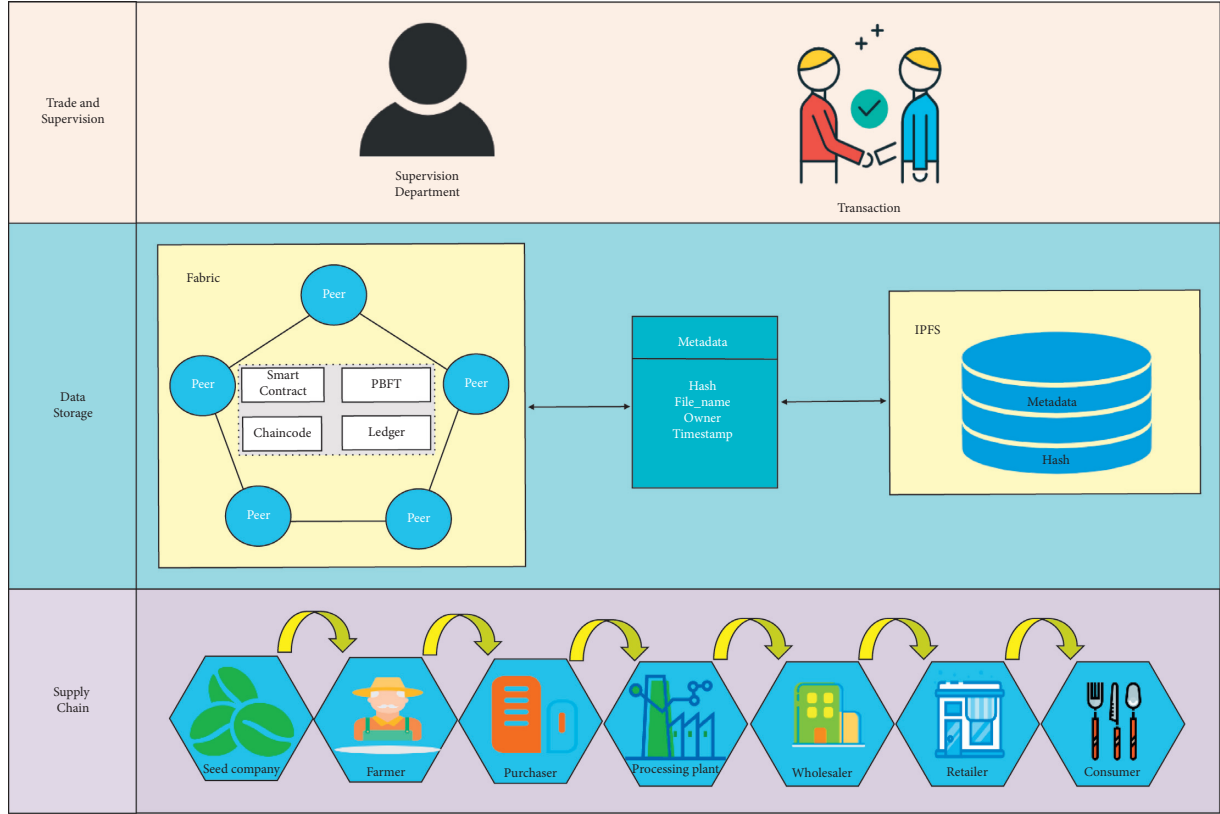


FIGURE 1: Agricultural product supply chain traceability solution based on Bc-IPFS.

TABLE 1: Symbol meaning.

Symbol	Meaning
ID	Identifier
H	Hash
Sig	User signature
PK	Public key
SK	Private key
m_R	Transaction order
\parallel	Connection symbol
ς	Ring signature
$Info$	Assessment information
$Value_{trust}$	Trust value
$Total_{trans}$	Total number of transactions

$$m_R = \Phi(m_t, \varsigma(SK_{sell}, m_t)). \quad (3)$$

Only after obtaining the completed order m_R , can the consumer release a comment on the product. Firstly, the consumer verifies the transaction signature $\varsigma(SK_{buy}, m_t)$ of the retailer

$$\Phi(mt, PK_{sell}, \varsigma(SK_{sell}, mt)) = 1. \quad (4)$$

Then, he/she verifies the seller's signature based on the seller's public key. Secondly, the consumer creates the ring signature ς based on the assessment information $Info = [ID_{pro} || H_{pro} || P_{sco} || P_{txt}]$ and sends the $(Info, \varsigma, m_R)$ to the blockchain. The blockchain verifies m_R and ς , and, upon successful verification, $Info$ will be stored in IPFS, and H_{Info}

will be stored in the blockchain network. In addition, the trust value

$$Value_{trust} = \frac{\sum(\alpha \cdot score_{ser} + \beta \cdot score_{qual})}{Total_{trans}}, \quad Total_{trans} \geq n, \quad (5)$$

can be calculated through consumers' evaluation of goods ($score_{ser}$), including service score and product quality score for retailers ($score_{qual}$), where the coefficient is $\alpha + \beta = 1 \forall \alpha, \beta \in (0, 1)$ and n refers to the number of transactions; even if individual consumers conduct malicious evaluation, the behavior will still have slight impact on the overall evaluation score, effectively reducing the negative effect of malicious evaluation on retailers [16, 17]. The total number of transactions must have at least n successful orders before the trust value is recognized, which can effectively protect new businesses from malicious comments at the initial stage. At the same time, only the first score of each natural month is valid for each user, thus avoiding malicious comments [18].

3.1.1. Data Storage/Query. Any data created from a transaction between both parties will be stored. As shown in Figure 2, first, the data will be sent by the http-post method, and when the predefined block size is achieved, the data will be partitioned, packed, and stored in IPFS, and the address of the IPFS storage block will be acquired. Then, the address will be stored in the Fabric blockchain, and the Fabric

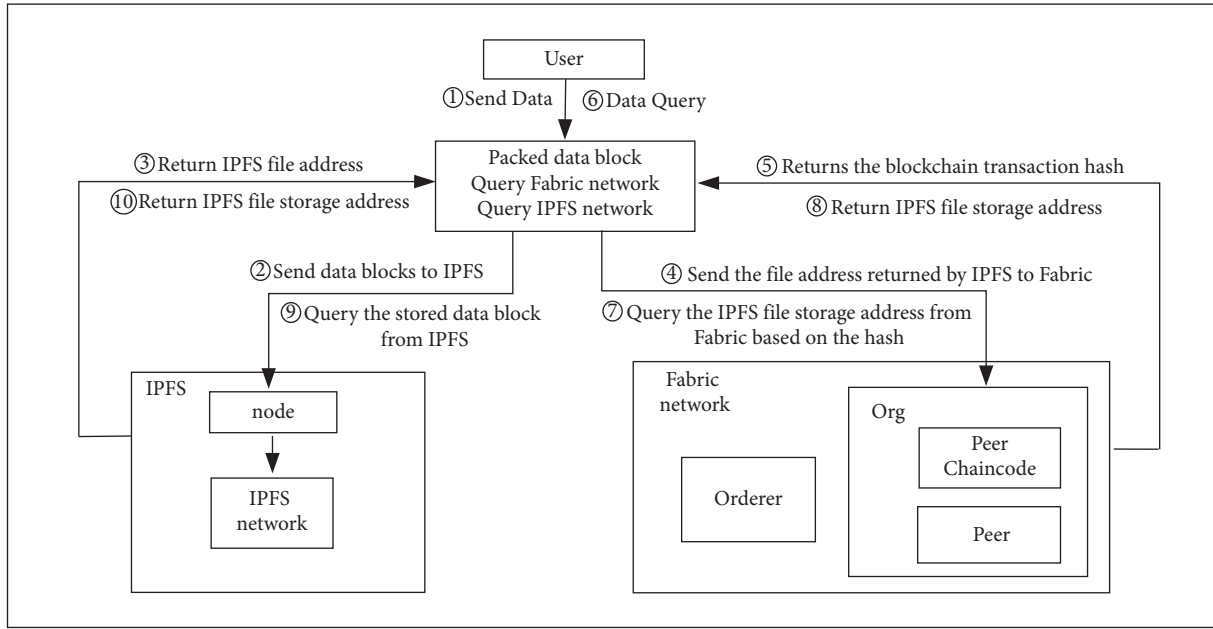


FIGURE 2: Functional design of data storage and query.

chaincode will be called to store the information in the peer node ledger for preservation. Upon data query, the http-get method will be adopted for data request, Fabric chain will be initiated to check whether the chaincode includes the IPFS address, and the transaction data will be retrieved from IPFS according to the hash address of the corresponding data block file on IPFS.

IPFS stores the data blocks of all source data, and the storage of data blocks is not subject to any sequence, but each data block is specifically correlated to the corresponding hash address, and the mapping relations between hash addresses and detail information is stored in the Fabric blockchain.

Nowadays, many problems of cloud storage resulted from improper server management and maintenance provided by decentralized cloud service providers or excessively centralized distribution of cloud service providers. If a file is stored in the cloud hard disks provided by the cloud service provider, and if the hard disks are put together in a centralized manner, even if the file is provided with the corresponding backup file, the hard disk in which the backup file is located may be stored in the hard disk in which the original file is located. As a result, the servers malfunction in case of power outage or other failures happens, and it cannot be accessed externally; the only way to the problem is to wait for recovery of the servers. But IPFS is not limited, as IPFS is a new type of Internet technology comparative to the HTTP protocol, solving the data storage and distribution problems, and it is designed to create permanent and decentralized storage and file-sharing methods through peer-to-peer network (PPN), with the concept of separating a file into many pieces scattered on different locations of the network, which can be acquired simultaneously from multiple servers upon downloading of the file. Even if certain servers are malfunctioning, it will not create adverse influence on the

access of external users to the entire network, nor on the users' data acquisition. In addition, even if certain node data is completely lost due to improper operation, there are many backups on the entire network. The advantages of IPFS are ideal to tackle the shortcomings of traditional centralized cloud storage, e.g., vulnerability to data leakage, vulnerability to hardware damage, and poor repair capacity.

In order to achieve large-scale distributed storage, there are three problems that need to be handled: (1) How to increase storage capacity, that is, to attract more users to provide storage resources, (2) how to improve retrieval efficiency and achieve rapid service response, and (3) how to guarantee that data storage and circulation are safe. In response to that, the researchers introduced blockchain technology as the incentive layer for distributed storage and obtained a series of research results. The most representative solution is a distributed storage system based on IPFS.

The storage party can prove its effective storage capacity through the proof of storage mechanism (Proof of Storage) to obtain tokens (the first problem) [19, 20]. Retrieval service parties can provide data retrieval services, and efficient retrieval can obtain more tokens (the second problem). The data security can be solved by encryption technology, and the blockchain can provide evidence of data access.

3.1.2. Relationship between Entity Sequences. As shown in Figure 3, the relationship between entities shows some key properties and functions of smart contract. The relationship between entities and smart contract is shown in Figure 3. Each participating entity in the supply chain participates by invoking the functions in the smart contract. Therefore, the metadata and relationships are of great importance to the realization of the smart contract [21]. The regulatory authority creates a master smart contract to be invoked by

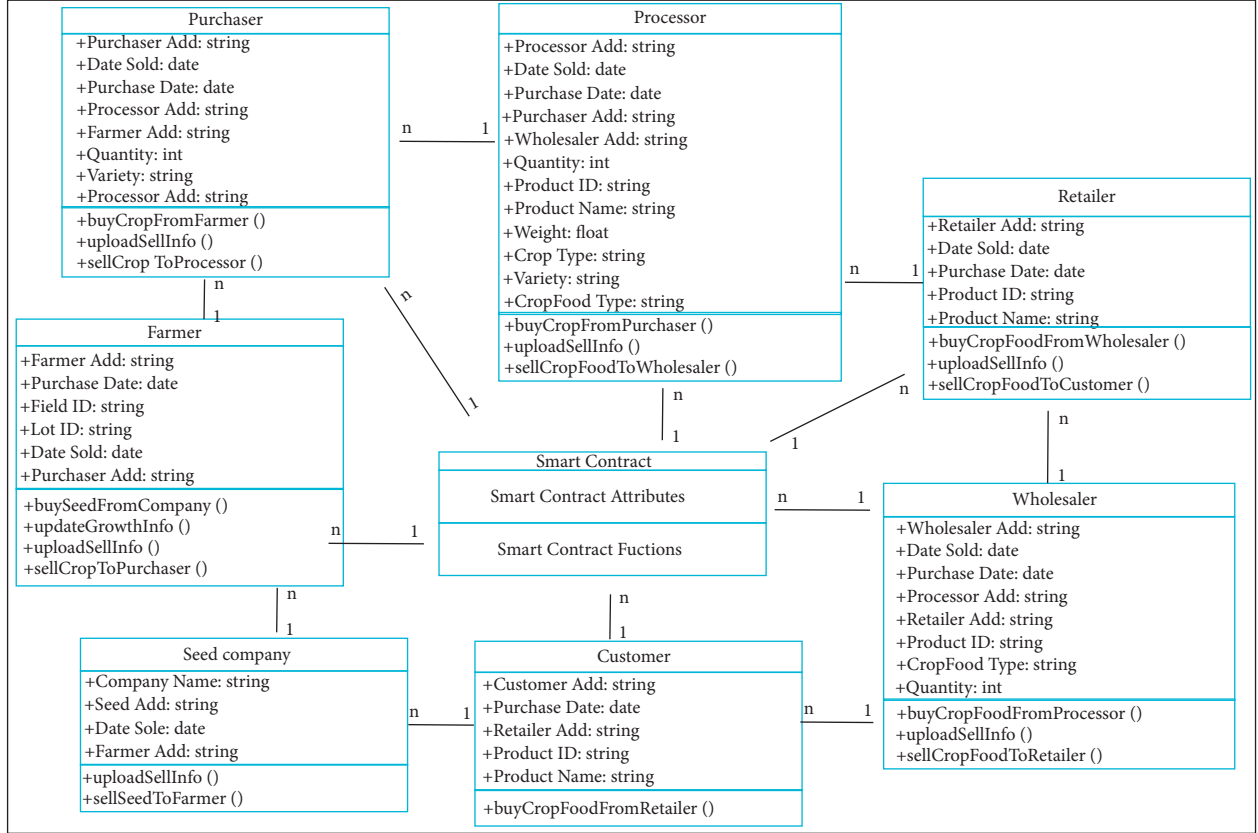


FIGURE 3: Entity relationship chart.

other entities. If certain changes happen to the relationship between the buyer and the seller, the corresponding contract will be executed. Once the parties have agreed on the details of the agreement, the transaction can proceed.

3.1.3. Algorithm. As mentioned above, the relevant supervision departments deploy the master smart contract, and each entity can invoke the trade through the interface. In the initialization state, each transaction entity needs to be registered; otherwise the transaction cannot be carried out. Later, the process of user registration, transaction, and product evaluation will be described in detail. Table 2 shows the interpretation of the variables in the algorithm.

3.2. User Registration. Algorithm 1: when users register their identities, the input parameters include ID_{user} , V_{IDuser} , $User_address$, and Pk_{user} . The algorithm is used for user registration, and the necessary user information is stored in the blockchain. Unregistered users cannot participate in the transaction. After the user registration verification is passed, the regulatory agencies call the smart contract interface `node_register()` to store user information in the blockchain.

3.3. Goods Registration. Algorithm 2: firstly, the goods owner registers the products so that the buyer can view the information of the goods he/she needs to buy and input the information of the goods to save, which plays a vital role in

traceability. The goods owner enters ID_{pro} , Pro_Lot , P_{typ} , P_{quan} , V_{IDpro} , ID_{own} , and PK_{own} .

3.3.1. Commodity Transaction. Algorithm 3 is jointly implemented by the buyer and the seller. Both parties negotiate P_{pri} , P_{quan} , and P_{qual} requirements of the goods. After reaching an agreement, the buyer pays for the goods Pay_pri . At the same time, the seller submits the deposit $Fine$, which is one half of the commodity transaction price λ , $\lambda \in (0, 1)$. The value of λ is decided by both parties, i.e.,

$$Fine = \lambda \cdot Pay_pri, \quad \lambda \in (0, 1), \quad (6)$$

$$Pay_pri = P_{pri} \times P_{quan} \& Fine = \lambda \cdot Pay_pri. \quad (7)$$

When formula (7) is satisfied, the smart contract triggers the function `Sell_agree()`. After the buyer confirms the receipt of the goods and the loan, the smart contract triggers `Pay_agree()` to pay the loan and deposit to the seller. If the buyer does not confirm the loan and there is no dispute, the transaction is considered to be successful within seven days, and the payment and deposit are also paid to the seller. When the transaction is done successfully, the smart contract calls `Trans_Record()` to record $address_buyer$ and $address_seller$, as well as the information of the traded goods. If the buyer has disputes on the goods and does not agree to pay or the buyer has disputes over the goods within seven days, the `Dispute_event()` will be triggered and should be

TABLE 2: Contract function description.

Function	Description
node_register()	Only the supervisory authority can call users whose registration has been successfully verified
node_record()	Only the supervisory authority can call it to record the object that this batch of goods belongs to at the time
release_pro()	This is used to release the information of the batch of goods
Trans_Record()	This is used to record transactions
Dispute_event()	Arbitrators or third-party agencies handle transaction disputes
Signature()	This is used for evaluation signature
Storage()	This is used to store records
Credit()	This is used to update reputation value

Input: ID_{user} , $V_{ID_{user}}$, $User_address$, PK_{user}

Output: Registration result

- (1) Users send ID_{user} , $V_{ID_{user}}$, $User_address$ and PK_{user} to regulatory agencies
- (2) Regulatory agencies verify users' identity information
- (3) **if** the verification is successful **then**
- (4) Regulatory agencies call node_register ($User_address$, PK_{user})//Only the regulatory agencies can call, register the user with successful verification, and send the relationship of $User_address$ and PK_{user} into address_pk.
- (5) Update address_pk
- (6) return "successfully registration"
- (7) **else**
- (8) **return** "registration failed. Please submit the real information to register again."
- (9) **end if**

ALGORITHM 1: User registration algorithm.

Input: ID_{pro} , Pro_Lot , P_{typ} , P_{quan} , $V_{ID_{pro}}$, ID_{own} , PK_{own}

Output: Registration result

- (1) Goods owner upload ID_{pro} , Pro_Lot , P_{typ} , $V_{ID_{pro}}$, ID_{own} onto the IPFS
- (2) Goods owner sends H_{pro} , $V_{ID_{pro}}$, PK_{own} to regulatory authorities
- (3) Regulatory agencies verify the owner of the goods
- (4) **if** the verification is successful **then**
- (5) Regulatory agencies call node_record (ID_{pro} , ID_{own} , Pro_Lot , $Time$)//It can only be called by regulatory agencies to record the owner of the commodity batch at that time
- (6) **return** "successful registration" **then**
- (7) Goods owner invokes release_pro (ID_{pro} , Pro_Lot , P_{typ} , P_{quan})//The goods are successfully registered by the owner, Later, the information of this batch of goods can be released for buyers to choose.
- (8) **else**
- (9) **return** "failed verification, please submit the information again"
- (10) **end if**

ALGORITHM 2: Goods registration algorithm.

handled by an arbitrator or a third-party organization. The place where the commodity is sold should be recorded for a single completed transaction, so that the commodity circulation record can be checked. In other words, both the buyer's and the seller's user address should be recorded in the transaction records, and these two addresses refer to the addresses registered in Algorithm 1.

3.4. Consumer Evaluation. In Algorithm 4, consumers can score the product and service quality of this transaction after an order completed and then upload such evaluation to IPFS.

4. Theory and Experiment Analysis

In this section, we analyze the traceability, security, and performance of the plan and make qualitative analysis by comparing this thesis with other papers at last.

4.1. Traceability Analysis. In this thesis, the solution of grain traceability of information is stored in IPFS files based on Hyperledger through chaincode (smart contract), in which transaction data is uploaded to IPFS to effectively solve the potential dangers such as high cost, waste of broadband, short text storage time, dependence on backbone network,

Input: $ID_{buyer}, ID_{seller}, address_{buyer}, address_{seller}, P_{pri}, P_{quan}, P_{quab}, Fine$
Output: Transaction result

- (1) The seller applies smart contract function: negotiate ($ID_{buyer}, ID_{seller}, P_{pri}, P_{quan}, P_{quab}$)//Both parties can negotiate the commodity price, quantity and quality through this function.
- (2) **if** $Pay_{pri} = P_{pri} \times P_{quan} || Fine = \lambda \cdot Pay_{pri}$ **then**
- (3) Contract status becomes Sell_agree
- (4) **if** Confirm receipt or time stamp > deadline **then**
- (5) Apply smart contract function: Trans_Record ($payment, address_{buyer}, address_{seller}$)
- (6) **else**
- (7) Trigger the contract event: Dispute_event ($ID_{buyer}, ID_{seller}, P_{pri}, P_{quan}, P_{quab}, Fine$)
- (8) **end if**
- (9) **else**
- (10) Contract status becomes Sell_disagree
- (11) **return** (“transaction failed”)
- (12) **end if**

ALGORITHM 3: Commodity trading algorithm.

Input: $ID_{seller}, m_R, \varsigma(SK_{buy}, m_t), PK_{sell}, \varsigma, Info$
Output: Evaluation agency verification results

- (1) The seller applies evaluation contract function: comment ($m_R, \varsigma(SK_{buy}, m_t), PK_{sell}$)
- (2) **if** $\Phi(m_t, PK_{sell}, \varsigma(SK_{sell}, m_t)) = 1$ **then**
- (3) Apply signature function: signature ($Info, \varsigma, m_R$)
- (4) **If** block verification passed **then**
- (5) Trigger contract storage function: storage ($Info, H_{Info}$)
- (6) Trigger contract credit update function: credit ($ID_{seller}, Value_{trust}$)
- (7) **else**
- (8) **return** (“verification failed”)
- (9) **end if**
- (10) **else**
- (11) **return** (“No review permission”)
- (12) **end if**

ALGORITHM 4: Consumer evaluation algorithm.

DDOS, XSS, and CSRF attack[22–25]. Meanwhile, combined with blockchain technology, it makes hash stored in blockchain to solve the limited data storage in blockchain efficiently and keep data stored safe and not tampered [26–28]. The information of each transaction involves the previous owner’s information before and after the transaction. The unique identifier and batch number of the grain are added to each subsequent transaction to form a complete traceability chain. When consumers give negative evaluation to the quality and safety of commodity, the regulatory authorities and retailers can trace the source quickly according to the relevant product batches in the evaluation, determine the production batches of the products, and locate the product batches for inspection in time. Consumers can also screen and select suitable purchase objects based on the retailer’s reputation value and product evaluation [29].

As shown in Figure 4, the access query records for the uploaded file are displayed. Each record contains the hash value, owner name, visitor, and time stamp of the accessed file.

4.2. Security Analysis

Unforgeability: distributed storage of data is allowed in IPFS, and the data will not be tampered and forged. The data stored in IPFS network cannot be altered without changing the data identifier. In IPFS, the identifier is an encrypted data hash [30]. It means that if the identifier of data is stored in the underlying distributed general ledger, noncritical data can be stored in IPFS. This can cut down consumption of operation in distributed ledger. Compared with centralized storage, if a hacker intercepts the request from hash and tries to send a malicious phishing site, the user can, with the help of data received through running hash function, compare the hash value of the data received with the one requested and reject the received data if not matched.

Consumer privacy: consumers make an evaluation through ring signature after obtaining m_R . The ring signature is anonymous, so attackers are not sure which ring member generates the signature. The probability is lower than $1/n$ even if the private key of ring member is acquired.

```

"File1":{"accessor":"33k5b3zd3CbhRnV8Qh7bYK9c4QGkp1VBcbXFSLSyce1aGT",
"filehash":"31Ra4FtkfZqD5EhpSeERAj4vy3LSkihKtLAH98SYRw9nRu",
"owner":"pikachu",
"time":"2021-10-13 14:18:34"},

"File2":{"accessor":"33WwsymTS93LbMWAbJYfE9LQnqwuRsN63E5iVpRN6PyMZA",
"filehash":"25qowBS4a8gjqMLm7FYhM6pRogy2T7TZc6gpxmpyqWPGqf",
"owner":"pikachu",
"time":"2021-10-13 14:21:12"},

"File3":{"accessor":"33Y3Vv4cVS4hTiyqyxfx3eX9BLQtR4NFojoC4Ct5MLjpd7",
"filehash":"21Ew3urZVj37WrPaiVdmnuJNsZg1MMYCLbFwSAPwQEHlAD",
"owner":"pikachu",
"time":"2021-10-13 14:27:37"},

"File4":{"accessor":"2xtBLHejVEeBBUCYFknxSvMnAarkNRqkFsZeybZtDsgrAA",
"filehash":"2z3hc1bDddRb1bMSDprzsZG25Zrjq8JTJU7MxWH1L7WLVdr",
"owner":"pikachu",
"time":"2021-10-13 14:35:42"},

"File5":{"accessor":"27F3PRTbRAe7aX1BMQqdB4uiYAACvaizFDKwmMtyVw4Jh",
"filehash":"272uTJpWcuZ4LxvonDKnwFPPpLcaB6MjqpAg6WuTYAmBz",
"owner":"pikachu",
"time":"2021-10-13 14:43:26"},
    
```

FIGURE 4: On-chain metadata for accessing files.

Transparency: both the consumer evaluation and the credit value of retailers are open to the public. Other consumers can screen these contents during commodity purchase. Meanwhile, the credit value can only be shown by setting a certain amount of turnover, effectively lowering the negative effect due to malicious evaluation from malicious consumers.

4.3. *Experimental Results and Analysis.* The test is done under the circumstance of Ubuntu 20.04 LTS on a computer equipped with Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz, running memory of 64 G, 1T mechanical hard disk, and 500 GB solid state hard disk. Ubuntu is built in mechanical hard disk. Generate a local IPFS node and obtain its public ID, create an IPFS network, and embed it in the Fabric blockchain, Hyperledger Caliper, an open source blockchain performance evaluation tool, is used to test transaction throughput and delay, and the test results are shown in Figures 5 and 6.

As shown in Figure 5, the query transaction and the R/W transaction show a bottleneck at the throughput of 25 tps and 22 tps, respectively. In case of low transaction rate, the submitted transaction must wait. Furthermore, transactions are continuously sent through the client, and it will show low delay of subsequent transactions received in each block timeout period, as shown in Figure 6. When the transaction rate is 10 tps, the query delay drops to about 0.75 tps, and if the transaction rate exceeds the maximum throughput, the accumulated transactions will show higher delay. Therefore, the delay continues to increase after the transaction rate exceeds 20 tps. The performance is related to network delay, consensus delay, chaincode execution time, block

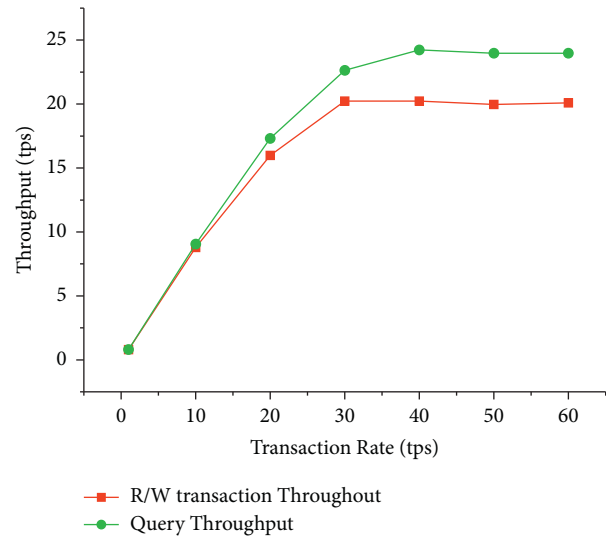


FIGURE 5: Transaction throughput.

verification delay, and other factors, but the system's throughput always has linear relation to the number of channels [31].

Since adding blockchain to IPFS will result in additional consumption of computing resources and time, thus this solution measures the performance variations of file reading when blockchain is adopted and when blockchain is not adopted. Thus, five files with the sizes of 0.5 T to 2.5 T are selected and uploaded to Ubuntu 20.04 LTS, data reading is conducted with the computers in the same network (Windows 10 , Intel(R) Core(TM) i7-4720HQ CPU @ 2.60 GHz, 12 GB RAM, and 500 GB Hard Disk), and the test results are shown in Figure 7. In the circumstance where

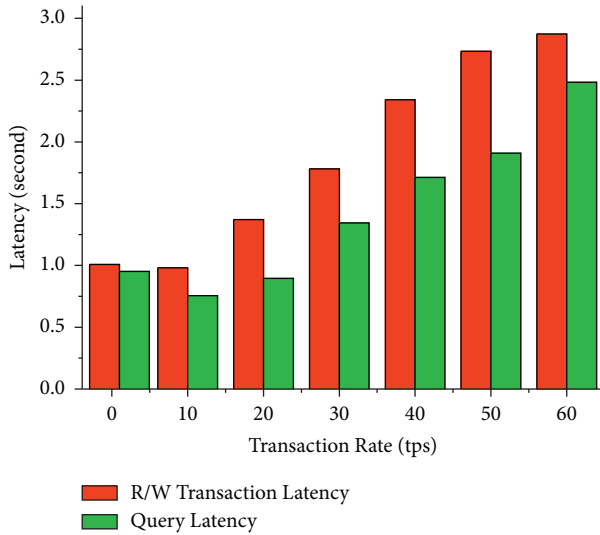


FIGURE 6: Transaction delay.

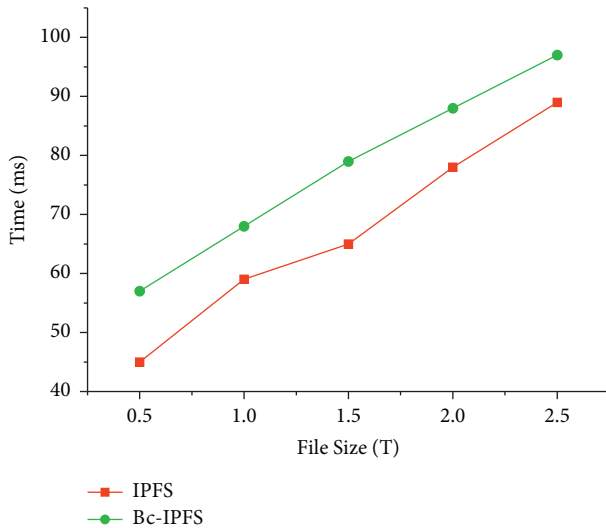


FIGURE 7: Compare the file read performance difference between IPFS and Bc-IPFS.

blockchain is adopted, the file reading is slightly lower than that of ordinary IPFS, as a blockchain transaction relies on the completion of IPFS processes, and the transaction time may also be adversely influenced by the local network speed and computer operation capacity. However, regardless of the sizes of the files added to IPFS, the sizes of the metadata stored on the blockchain do not show any signs of evident changes, as shown in Figure 8.

4.4. Scheme Comparison. First of all, Hyperledger is a private blockchain technology, distinctive from the blockchain technology such as ETH and Bitcoin, and the members of a blockchain network are known to each other, and the membership is available to the public, allowing new members to join in and carry out transactions on the network [32]. Hyperledger is a type of blockchain managed by

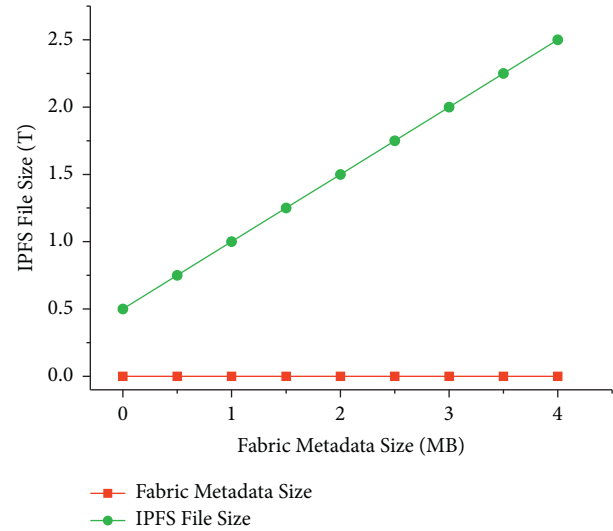


FIGURE 8: Comparing the size of files on Bc-IPFS and the size of file metadata on blockchain ledger.

multiple organizations or institutions, and its data can only be read, written, and maintained by those organizations or institutions. It effectively tackles the problem of “information silo” between different entities, which is ideal to the security traceability systems of agricultural products. The specific reasons of why Hyperledger Fabric is more adaptable to the solution proposed in this thesis, compared with ETH, are as follows:

- (1) *Expandability.* One of the main characteristics distinguishing Hyperledger Fabric from other blockchain technologies such as ETH is its modularity, which provides an architectural structure of modularity and expandability applicable in various environments, making it more adaptable to the functional expansion of IPFS [33].
- (2) *Consensus Mechanism.* The current ETH adopts the Proof of Work (POW) consensus mechanism, which is inefficient in accounting and vulnerable to 51% hash rate attack, and it consumes a great deal of computer resources. The Hyperledger adopts the Byzantine Fault Tolerance (PBFT) consensus mechanism, which provides an accounting efficiency at the sec level and low power consumption; thus it is ideal for the development of the traceability industry [34].
- (3) *Confidentiality.* Since ETH is a public network irrelevant to the concept of authority and is completely transparent, thus all transactions recorded on the blockchain network can be available to and accessible by each counterparty, but Hyperledger is a blockchain platform with access authority and high levels of security; thus all transactions are only available to the ones with access authority.
- (4) *Interactivity.* Hyperledger Fabric provides SDKs for interaction, so it can interact with IPFS and can effectively search the blockchain and provide data for review through the functions provided by SDKs.

TABLE 3: Plan comparison.

Characteristic	[5]	[7]	[10]	[35]	This work
Traceability	Yes	Yes	Yes	Yes	Yes
Supervisory	No	Yes	No	Yes	Yes
Scalability	No	Yes	Yes	Yes	Yes
Evaluation	No	No	No	No	Yes
Gas	—	No	Yes	Yes	No
On-chain data	High	Low	Low	High	Low

Then, we compare the traceability plan designed here with other plans, and the results are shown in Table 3 [7]. The data is still stored centrally, although traceability is based on blockchain. In contrast, this plan has higher decentralization. In addition, in this plan, a supervision organization is set up to supervise the members of the supply chain and products, ensure the integrity and accuracy of information, and strengthen the supervision of the organization [10]. Obviously, relevant entities in the supply chain are not supervised efficiently. At the same time, the consumer evaluation function designed in this plan plays a strong role in retailers' self-monitoring, and, through retailers' credit value function, consumers are able to quickly screen the purchase objects so that it can effectively reduce the negative effects due to malicious evaluation behavior. K. Salah [35] solves the problem studied by Ethereum smart contract. This method needs to cost the Gas fee. Once the Gas is used up, the contract will not be executed and the used fee will not be refunded. However, Hyperledger is not involved in mining process, which can save resources and time, and has modularity and expansibility.

5. Conclusion

The Bc-IPFS-based solution proposed in this thesis tackles the problem of "information silo" between entities by the alliance blockchain Hyperledger and realizes the automation of traceability through the smart contract deployed on Hyperledger Fabric, so as to improve efficiency. The IPFS technology is adopted to ensure data security and overcome the data storage constraint of blockchain for overwhelming data, the ring signature algorithm is adopted to privatize consumer information and encourage consumers to timely feedback product problems, and performance evaluation is conducted by virtue of throughput capacity and delay, in addition to security analysis of the solution. The solution is developed with comprehensive considerations to traceability, transaction, and retailer reputation, and the established reputation function can be used to maintain the reputation of the entities in the supply chains of agricultural products and the quality rating of the products. This thesis describes the plan design, overall structure, entity relationship diagram, interaction, and details related to implementation algorithm and shows how to apply the plan to track grain supply chain. The plan designed can provide and meet reliable decentralized traceability demands for any crop in agricultural supply chain.

Up to now, the system based on blockchain is still challenged by its practical implementation. In the future, we plan to integrate protection of enterprise and consumer privacy in agricultural food trade. Similarly, the retailer's credit value comes from consumers' evaluation, which may be biased or falsified. As a result, we plan to design a supervision mechanism for comment information to help improve such accuracy.

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 sponsored by the National Natural Science Foundation of China under grant no. 62172353. Future Network Scientific Research Fund (project no. FNSRFP-2021-YB-48), Science and Technology Program of Yangzhou City (no. YZU202003). Natural Science Foundation of the Jiangsu Higher Education Institutions (grant no. 17KJB5 20044), and Six Talent Peaks Project in Jiangsu Province (no. XYDXX-108).

References

- [1] X. Yuan, "Research on the credit and law enforcement mechanism of food safety in my country," *Food Safety Guide*, pp. 32-33, 2016.
- [2] L. Zhang, M. Peng, W. Wang, Y. Su, S. Cui, and S. Kim, "Secure and efficient data storage and sharing scheme based on double blockchain," *CMC-Computers Materials & Continua*, vol. 66, pp. 499-515, 2021.
- [3] M. Crosby, "Blockchain technology: beyond bitcoin," *Applied Innovation*, vol. 2, p. 71, 2016.
- [4] W. Yu and S. Huang, "Traceability of food safety based on block chain and RFID technology," in *Proceedings of the 2018 11th International Symposium on Computational Intelligence and Design*, pp. 339-342, Hangzhou, China, December 2018.
- [5] F. Tian, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," in *Proceedings of the 2016 13th International Conference on Service Systems and Service Management*, pp. 1-6, Kunming, China, June 2016.
- [6] F. Tian, "A supply chain traceability system for food safety based on HACCP blockchain & Internet of Things," in *Proceedings of the 2017 International Conference on Service*

- Systems and Service Management*, pp. 1–6, Dalian, China, Jun. 2017.
- [7] X. Yang, M. Li, H. Yu, M. Wang, D. Xu, and C. Sun, “A trusted blockchain-based traceability system for fruit and vegetable Agricultural products,” *IEEE Access*, vol. 9, pp. 36282–36293, 2021.
 - [8] C. Xie, Y. Sun, and H. Luo, “Secured data storage scheme based on block chain for agricultural products tracking,” in *Proceedings of the 2017 3rd International Conference on Big Data Computing and Communications*, pp. 45–50, Chengdu, China, August 2017.
 - [9] J. T. Hao, Y. Sun, and H. Luo, “A safe and efficient storage scheme based on blockchain and IPFS for agricultural products tracking,” *Journal of Computers*, vol. 29, no. 6, pp. 158–167, 2018.
 - [10] K. Salah, N. Nizamuddin, R. Jayaraman, and M. Omar, “Blockchain-based soybean traceability in agricultural supply chain,” *IEEE Access*, vol. 7, pp. 73295–73305, 2019.
 - [11] E. Nyaletey, R. M. Parizi, Q. Zhang, and K.-K. R. Choo, “BlockIPFS - blockchain-enabled interplanetary file system for forensic and trusted data traceability,” in *Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 18–25, Atlanta, GA, USA, July 2019.
 - [12] Hyperledger Fabric Documentation: https://hyperledger-fabric.readthedocs.io/zh_CN/latest/whatis.html, March 20 2021.
 - [13] A. Lohachab, S. Garg, B. H. Kang, and M. B. Amin, “Performance evaluation of Hyperledger Fabric-enabled framework for pervasive peer-to-peer energy trading in smart Cyber-Physical Systems,” *Future Generation Computer Systems*, vol. 118, pp. 392–416, 2021.
 - [14] 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.
 - [15] M. Kumar and S. Chand, “MedHypChain: a patient-centered interoperability hyperledger-based medical healthcare system: regulation in COVID-19 pandemic,” *Journal of Network and Computer Applications*, vol. 179, Article ID 102975, 2021.
 - [16] X. Xue, Z. Chen, S. Wang, Z. Feng, Y. Duan, and Z. Zhou, “Value entropy: a systematic evaluation model of service ecosystem evolution,” *IEEE Transactions on Services Computing*, p. 1, 2020.
 - [17] X. Xue, S. Wang, L. Zhang, Z. Feng, and Y. Guo, “Social learning evolution (SLE): computational experiment-based modeling framework of social manufacturing,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3343–3355, 2019.
 - [18] S. Su, Z. Tian, S. Liang, S. Li, S. Du, and N. Guizani, “A reputation management scheme for efficient malicious vehicle identification over 5G networks,” *IEEE Wireless Communications*, vol. 27, no. 3, pp. 46–52, June 2020.
 - [19] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, “Semi-selfish mining based on hidden Markov decision process,” *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.
 - [20] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, “Is semi-selfish mining available without being detected?” *International Journal of Intelligent Systems*, 2021.
 - [21] L. J. Zhang, Z.-D. Liu, X. Guo, and X. Xiao, “Secure data sharing model based on smart contract with integrated credit evaluation,” *Acta Automatica Sinica*, vol. 47, no. 3, pp. 594–608, 2021.
 - [22] R. Kumar and R. Tripathi, “Towards design and implementation of security and privacy framework for Internet of Medical Things (IoMT) by leveraging blockchain and IPFS technology,” *The Journal of Supercomputing*, vol. 77, pp. 1–40, 2021.
 - [23] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, “PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs,” *International Journal of Intelligent Systems*, 2021.
 - [24] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, “CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine-learning techniques,” *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2021.
 - [25] M. Shafiq, Z. Tian, A. Bashir, X. Du, and M. Guizani, “IoT malicious traffic identification using wrapper-based feature selection mechanisms,” *Computers & Security*, vol. 94, Article ID 101863, 2020.
 - [26] R. Kumar, R. Tripathi, N. Marchang, G. Srivastava, T. R. Gadekallu, and N. N. Xiong, “A secured distributed detection system based on IPFS and blockchain for industrial image and video data security,” *Journal of Parallel and Distributed Computing*, vol. 152, pp. 128–143, 2021.
 - [27] M. Shafiq, Z. Tian, A. A. Bashir, A. Jolfaei, and X. Yu, “Data mining and machine learning methods for sustainable smart cities traffic classification: a survey,” *Sustainable Cities and Society*, vol. 60, 2020.
 - [28] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, “Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city,” *Future Generation Computer Systems*, vol. 107, pp. 433–442, 2020.
 - [29] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, “A survey on access control in the age of internet of things,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4682–4696, June 2020.
 - [30] E. Politou, E. Alepis, C. Patsakis, F. Casino, and M. Alazab, “Delegated content erasure in ipfs,” *Future Generation Computer Systems*, vol. 112, pp. 956–964, 2020.
 - [31] P. Thakkar, S. Nathan, and B. Viswanathan, “Performance benchmarking and optimizing hyperledger fabric blockchain platform,” in *Proceedings of the 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp. 264–276, Milwaukee, WI, USA, September 2018.
 - [32] T. Sato and Y. Himura, “Smart-contract based system operations for permissioned blockchain,” in *Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–6, Paris, France, February 2018.
 - [33] Hyperledger Blockchain Performance Metrics: 2021, https://www.hyperledger.org/wp-content/uploads/2018/10/HL_Whitepaper_Metrics_PDF_V1.01.pdf.
 - [34] L. Zhang, Y. Zou, W. Wang, Z. Jin, Y. Su, and H. Chen, “Resource allocation and trust computing for blockchain-enabled edge computing system,” *Computers & Security*, vol. 105, Article ID 102249, 2021.
 - [35] S. Wang, D. Li, Y. Zhang, and J. Chen, “Smart contract-based product traceability system in the supply chain scenario,” *IEEE Access*, vol. 7, pp. 115122–115133, 2019.