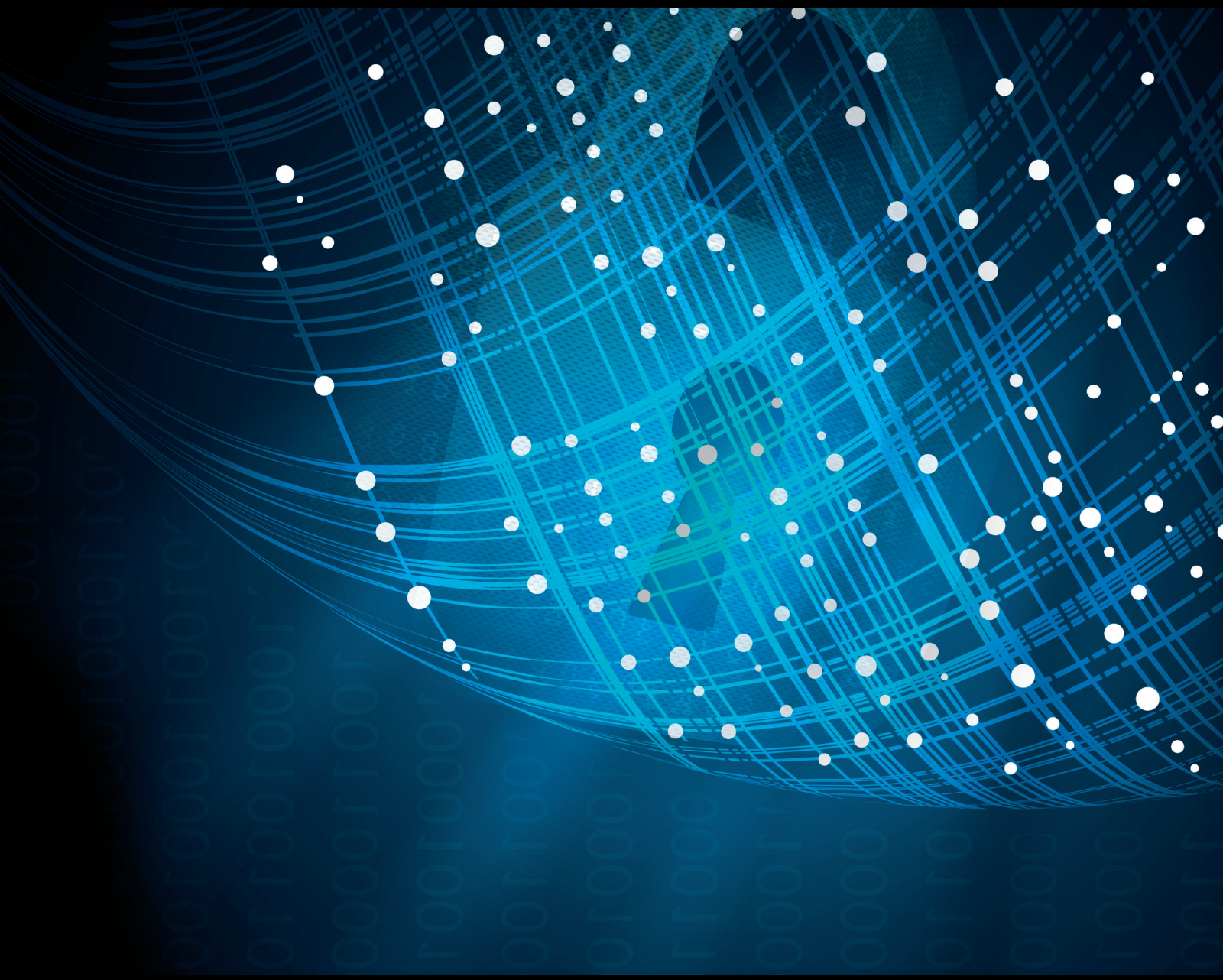
Security and Communication Networks

Secure Data Outsourcing in Blockchain-Based Internet of Things

Lead Guest Editor: Yinbin Miao
Guest Editors: Kim-Kwang Raymond Choo, Ximeng Liu, and Zhiquan Liu



Secure Data Outsourcing in Blockchain-Based Internet of Things

Security and Communication Networks

Secure Data Outsourcing in Blockchain-Based Internet of Things

Lead Guest Editor: Yinbin Miao
Guest Editors: Kim-Kwang Raymond Choo, Ximeng Liu, and Zhiquan Liu

Contents

Hindawi
Security and Communication Networks
Volume 2023, Article ID 6403686, 10 pages
https://doi.org/10.1155/2023/6403686

Research Article

Contract-Based Incentive Mechanism for Redactable Proof-of-Stake Blockchains

Yumei Wang, Yongdong Wu, and Junzuo Lai

Jinan University, Guangzhou 510632, China

Correspondence should be addressed to Yongdong Wu; wuyd007@qq.com

Received 21 October 2022; Revised 7 April 2023; Accepted 3 May 2023; Published 17 May 2023

Academic Editor: Andrea Michienzi

Blockchain has received a lot of attention due to its immutability. However, the immutability characteristic prohibits editing the blocks which need to be modified. Although the existing redactable blockchain enables to manipulate blocks in a controlled way, it may suffer from the security threats if the number of honest committee members (CMs) is insufficient. Thus, to attract honest CMs for validating and voting the editing blocks in permissionless blockchain, this paper presents a contract-based incentive mechanism between contract issuer and every CM. Firstly, it models the interaction between the contract issuer and each CM in the verifying and voting process. Secondly, it builds an incentive mechanism according to the contract issuer's cost and the committee size. Finally, it selects a sufficiently large number of CMs with an optimization method. The analysis shows that the present mechanism is secure against Sybil attack, and the simulations demonstrate that the proposed mechanism is effective.

1. Introduction

As the underlying technology of Bitcoin proposed in 2008 [1], blockchain has received widespread attention due to its immutability merits. Nevertheless, the immutability of blockchain has shown some side effects. For example, if a blockchain has been misused to store and distribute inappropriate content such as child pornography and material that infringes on intellectual property rights on the chain, the immutability of the blockchain prevents fulfilling the data regulations such as the "right to be forgotten" [2] and General Data Protection Regulation (GDPR) [3]. As a result, some chain participants may be reluctant to participate the blockchain for fear of being accused of possessing illegal information.

To overcome the shortcoming of the immutability, a redacting process is employed to rewrite the block data in a secure and controlled manner [4]. Usually, it varies with the consensus mechanisms in the blockchains. For the most popular POW blockchains and POS blockchains, cryptographic primitive based redacting and voting based redacting are preferable, respectively.

As a POS-like blockchain consumes much less energy than a POW counterpart and is deployed widely, this paper focuses on the POS blockchain, in particular to its redacting process. Generally speaking, a redacting process for POS blockchain composes of four main steps (e.g., [5]): (1) submitting an editing request from a user; (2) selecting a leader and Committee Members (CMs) in blockchain; (3) voting on the editing request from CMs; and (4) updating the block data. Specifically, at the beginning of each editing time slot, CMs are pseudo-randomly selected as volunteers according to their stakes by a verifiable random function (VRF) [6] and then verify the editing blocks before deciding whether to vote on them. However, if CMs spend resources to vote without reward, they may be reluctant to participate honestly in validating editing blocks and voting on candidate blocks over time. As a result, it increases the security risk that the data on the chain are maliciously tampered with.

In the editing process, it is critical to select as many honest CMs as possible to reduce security risks because rewriting old consensus blocks requires a stricter consensus approach. We believe that the higher the voting power, the more honest the CMs will be, because the larger the

percentage of stake in the blockchain, the less they want the blockchain to suffer from the security risk of tampering. To this end, this paper designs an incentive mechanism to motivate more CMs with higher voting rights to join in the validation and voting of rewriting blocks. Thus, it has to address two challenging problems. Firstly, the leader does not know in advance which stakeholder would become a CM and would be willing to participate in validation and voting. Secondly, he does not have an accurate value of the CMs' voting rights and does not know how the CMs would vote. Information asymmetry between the leader and CMs may lead to high costs for the leader to complete the editing process. Therefore, the best strategy for a leader is to design an incentive mechanism that reduces the impact of information asymmetry. In addition, the more CMs contribute, the more rewards they will receive. Accordingly, this paper presents a contract-based incentive mechanism. The addition of contracts allows the scheme to not only effectively motivate CMs to participate in redactable block validation and voting but also to maximise the utility of the leader. The contributions of this paper can be summarised as follows:

(1) Design an incentive mechanism to inspire more CMs with higher voting power to participate honestly in validating edit blocks and voting on candidate blocks. As long as a CM completes her validation and voting tasks, she will be rewarded with a portion of the transaction fee provided by the leader.

(2) Propose an enhanced redactable POS blockchain scheme for permissionless systems, so as to mitigate the security risk of tampering with data on the chain. The security analysis shows that the present scheme is secure against Sybil attack.

(3) Carry on abundant simulations to demonstrate that the present contract-based incentive mechanism achieves high performance in member utility compared to a contract with no information asymmetry. Thus, the present mechanism will have honest CMs enough to engage in voting.

The rest of the paper is organized as follows. Section 2 introduces the related work in redactable blockchain and incentive mechanism for blockchain. Then, in Section 3, we introduce the system model and the attack model. In Section 4, we introduce the overview of the enhanced Redactable POS Blockchain. The problem formulation and optimal contract designing for information asymmetry are elaborated in Section 5. Section 6 evaluates the performance of the designed contract. Finally, Section 7 concludes this paper and presents future work.

2. Related Work

Nowadays, the growing fusion of blockchain technology with other fields has been contributed by many scholars. Rathee et al. [7] proposed a blockchain framework that addresses the security problem of malicious intrusion on smart devices by adversaries in the Internet of Vehicles. Further, Rathee et al. [8] proposed a device-trustworthy management approach with the help of blockchain-based data transparency for the possible network adversaries in industrial Internet of things (IoT). Krishnamurthy et al. [9] proposed a voting layout based on blockchain and IoT devices in order to enhance the security of e-voting. In addition, Cai et al. [10] proposed an oracle protocol by utilizing alternative mechanisms to filter objective information from subjective data. The expanding applications of blockchain have also led to increased concerns about the security of data on the chain. Therefore, this section briefs the redactable scheme and incentive mechanism. The incentive mechanism is used to attract the CM so as to guarantee the security of the redactable scheme.

2.1. Redactable Scheme. To remove harmful data in the blockchain, redactable blockchain has been proposed. According to the authorization and modification method, the existing redactable schemes are mainly divided into two types: authorization-based chameleon hashing function and voting-based double hash chain.

In the redactable scheme with authorization-based chameleon hashing function, the trapdoor of the chameleon hash function is used to calculate hash collisions for arbitrary input data, thus enabling changes to block data without changing the original block connection. Ateniese et al. [4] first proposed a block-level redactable scheme based on chameleon hash functions, where authorized entities can obtain trapdoors and compute hash collisions for the corresponding blocks. Further, Derler et al. [11] proposed the policy-based chameleon hashes (PCH), which refers to the ability of anyone with all the permissions required by a policy to have the ability to compute arbitrary collisions for a given hash and hence enables fine-grained and controlled editing at the transaction level. Subsequently, accountability [12], revocation [13], supervision [14], and k-time [15] are embedded to make the editable scheme be more relevant to practical applications.

In the voting-based redactable scheme, anyone who harvests enough votes will be able to reach a consensus among the users on the chain to change the block. In order to eliminate a trusted central authority, Deuber et al. [16] proposed a block-level double hash chain scheme under nonauthorisation through consensus-based voting. This scheme extends the structure of adjacent blocks by preserving a copy of the Merkle root in its original state. In such a way, the integrity of the hash link among blocks is not broken, even if the hash value of the new block changes. To speed up voting for consensus, Li et al. [5] proposed an instantly editable blockchain protocol for POS and POW. The protocol pseudo-randomly selects the committee based on stake or computing power. That committee will validate edit blocks and voting on candidate blocks. Of these, completing a redaction requires only a time slot in the case of a synchronous network in POS blockchain.

In general, most editing schemes based on chameleon hash function require a trusted central entity to grant editing rights, and some schemes still require complex multiparty

computation (MPC) to manage chameleon hash traps, and the nondisclosure of traps makes it impossible for the public to verify the edited blocks. The voting-based consensus editing schemes achieve a decentralized, publicly verifiable editing process and do not require complex cryptographic primitives. However, the voting-based schemes demand a high level of honesty from the members involved in validating the editing blocks. In practical scenarios, often rational members are unwilling to spend extra computational resources and time to participate in the editing process. Therefore, this paper investigates an incentive mechanism to motivate members to honestly participate in the editable voting process.

2.2. Incentive Mechanism. As POS-type blockchain becomes more and more popular, Kang et al. [17] built a Stackelberg game to jointly maximise the utility of blockchain user and the profit of each miner on the POS-based consortium blockchain network. The game is designed to incentivize miners to participate in the verification and propagation of mined block, using the transaction fee of the blockchain user as a reward. However, it may be not practical because the game model assumes the information between leader and CMs is symmetrical. Later, Kang et al. [18] proposed a delegated proof of stake (DPoS) blockchain. The scheme is designed to allow highly reputable candidates to be selected as active miners and standby miners. Incentive is designed to motivate candidate miners to participate in block validation and prevent internal collusion among active miners. The designed mechanism is based on contract theory with asymmetric information. However, it is not applicable to secure data editing in the POS blockchain.

In summary, contract theory and other game theoretic approaches have been applied and developed in the blockchain domain. Nevertheless, there is a common problem with many of these works: the universality of the methods is not high. Consequently, in order to be able to effectively solve the incentive problem in the editable blockchain scenario, further research on secure and feasible contract theory schemes is necessary.

3. Preliminaries

In this section, we introduce the system model and the security model considered in this paper.

3.1. System Model. The current voting-based editable blockchain [5] retains the data structure of the block header and block body with a new replica of the Merkle root of the original data in the block header. This double hash chain model ensures the integrity of the blockchain after data modification. To the best of our knowledge, there is a lack of incentive measures for committee in existing voting-based editable blockchain, i.e., committees are not rewarded for their contribution. This greatly reduces the interest of CMs in participating in data editing. Therefore, in this paper, we propose an editable scheme with incentives that allows CMs to actively choose whether to complete voting and validation tasks based on workload and rewards.

At the high level, our system consists of three entities as shown in Figure 1: leader, user, and CM. When there is no user request to edit, the blockchain elects the leader via an underlying POS-based protocol, which generates the next block as usual. Otherwise, a committee is elected locally using the VRF to make pseudo-random decisions based on the stakeholder's stake. The committee needs to participate in the vote on the edit request. The output of VRF and a staked cryptographic sortition method will determine how many votes the member will get. The number of votes an edit block receives above a certain threshold is considered a consensus reached by the whole network. The specific design of the above process can be found in [5]. However, CM requires some computational resources and power to validate the edit block and complete the voting process. To encourage nodes to participate in the block editing process, our designed system rewards participating members in the form of transaction fees. More details about the scheme are given in Section 4. And we introduce the problem formulation, optimal contract in Section 5. For the convenience of the readers, we have listed the main notations used in the paper in Table 1.

3.2. Security Model. With reference to Figure 1, the participants jointly update a permissionless redactable blockchain. The blockchain stores the edited blocks and nonedited blocks including transactions submitted by the user, validated and voted by the CMs, and recorded by the leader. The committee that votes on the editable blocks is elected based on the stakes among stakeholders. A leader is supposed to be honest and rational in the present protocols.

A user is assumed to be malicious if he behaves in the following ways: (1) broadcasts a large number of meaningless edit requests and (2) publishes an edit request intended to add harmful data to the chain.

A CM can be honest, lazy, or malicious. In particular, (1) members are considered honest when they have truly validated the editable block and voted honestly based on the results of the validation; (2) they are assumed to be lazy when they skip the editable block validation or vote for it randomly; (3) malicious members vote in the opposite way in accordance with the validation results. The tolerance for malicious CMs is strictly less than $(T/2)$, where T is the expected committee size of stakes. A malicious vote by a CM could result in illegally tampering with the data on the chain.

4. Overview of the Enhanced Redactable POS Blockchain

In this section, a contract-based redactable POS Blockchain is described, where the contract is designed as an incentive mechanism to motivate higher voting rights committees to join the validation and voting of editable blockchain data. In other words, in order to ensure the consistence among all the nodes in the blockchain after editing, the number of honest CMs shall be sufficiently high in the validation and voting of

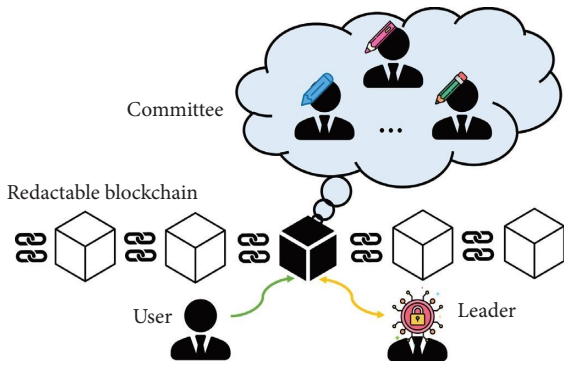


Figure 1: The system model.

Table 1: List of main notations.

Notations	Description
$\mathcal{U}$, G	Set of CMs, types of CMs be classified
θ_g	g-type
R_g, F_g	g-type CMs' incentives, g-type CMs' resources
U_l, U_m	Utility of leader, utility of CM
$c^{'}$	Unit value of resources for leader
c	Unit cost of computing resources spent by a CM
P_g	Prior probability of a g-type CM
$v(\bullet)$	A monotonically increasing valuation function
$R_{\max}$	Total editable transactions' fee
UC	Utility of a CM for launching the Sybil attack

editing block. Hence, it is in desire to develop an incentive mechanism to attract CMs.

As illustrated in Figure 2, we present an overview of the scheme to enhance the security of the redactable blockchain by embedding a contract-based incentive mechanism. A detailed design of the mechanism will be described in Section 5. The present redactable POS blockchain has three entities: (1) blockchain users, (2) leader, and (3) CMs. If a user wants to make an edit request and after broadcasting that request to the network, the mechanism allows for the editing of blocks by performing the following main steps:

① The leader develops the contract set based on the information he can gather about the committee and then broadcasts the set of contracts to the blockchain network.

② Each CM selects a corresponding contract and signs it and then carries out the tasks in accordance with the provisions of the contract.

③ The CM returns a proof including the voting outputs (i.e., edited blocks' verification and voting results) to the leader.

④ The leader gathers the proofs to verify the eligibility of voters and the vote results. When the number of the proofs associated with the editing blocks is higher than the vote threshold, the proofs are compressed and packed into a new block. Afterwards, a fee is paid to the corresponding CMs in accordance with the terms of the contract.

Finally, similar to [5], blockchain users check that whether the edit blocks meet the policy, i.e., whether the votes exceed the threshold and whether the blocks embedded in the votes satisfy the requirements of the blockchain. If yes, blockchain users will update the data locally.

5. Contract-Based Incentive Mechanism

Section 4 describes the overview of the enhanced redactable blockchain which includes an important incentive mechanism for attracting CMs. This section will elaborate the mechanism.

5.1. Problem Formulation. In each time slot, a monopoly market consists of a leader as task issuer and a set of CMs $\mathcal{U}$. In order to attract more CMs with high voting rights in validation and voting, we use the level of votes as a classification criterion for the type of CMs, i.e., CMs can be classified according to their votes: $\theta_1 < \ldots < \theta_g < \ldots < \theta_G$, $g \in \{1, \cdots, G\}$, where θ_g denotes a CM with a number of votes within a certain range. In this paper, assume that all CMs are rational. That is to say, a CM with more votes pays more attention to take part in the voting process.

The leader must overcome the resulting economic loss due to the information asymmetry caused by the leader not knowing the specific types of CMs. The leader offers CMs of different types contracts containing a series of reward-performance packages $(R_g(F_g), F_g)$. In this case, F_g is the validation and voting performance requirement for a CM of θ_g, and $R_g(F_g)$ is the corresponding reward to a CM of θ_g. If a CM completes a validation and voting task to a higher quality, i.e., the more computing resources invested, the more rewards the member will receive.

(1) Utility of the Leader: Depending on the contract (R_g, F_g) between a CM of type g and the leader, the utility function of the leader can be expressed as follows:

$$U_l(\theta_g) = c^{'} F_g - R_g, \tag{1}$$

where $c^{'} > 0$ is the unit value of computing resources for the leader, F_g is the required computing resources provided to the leader by a CM of type g, and R_g is the reward that leader must provide to a CM of type g under the contract (R_g, F_g) where the reward refers to the transaction fee provided by the blockchain users who make the edit requests. The utility of the leader is the benefit generated from the resources invested by the CMs minus the incentive to the CMs. For a validation and voting task with G types of CMs participating, the total utility available to the leader as task issuer is

$$U_l = \sum_{g=1}^{G} \left(|\mathcal{U}|P_g\right)\left(c^{'} F_g - R_g\right), \tag{2}$$

where P_g is the prior probability of a CM of type g, and $\sum_{g=1}^{G} P_g = 1$. According to reference [5], it is known that the ballot of a CM is broadcast to the entire blockchain network, where the ballot contains

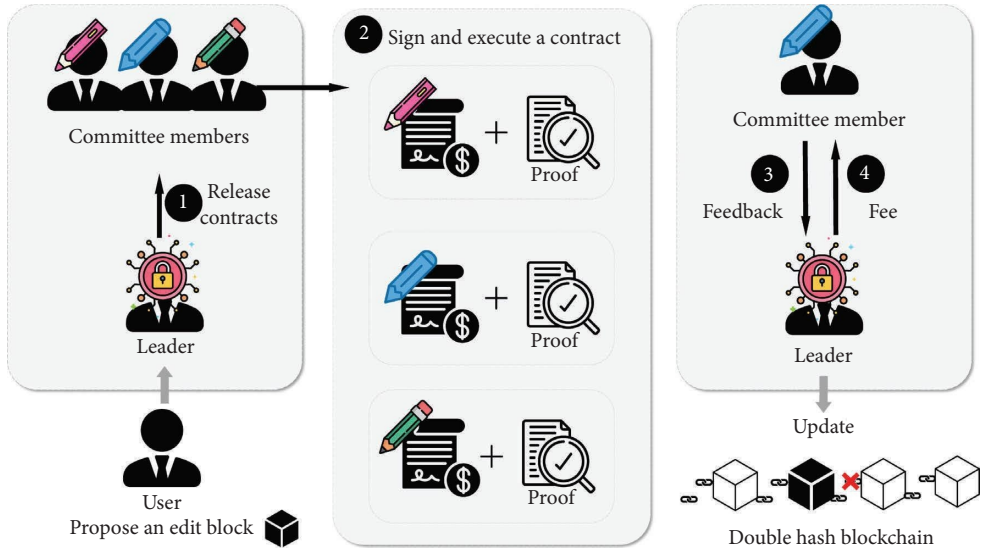


Figure 2: Diagram of the enhanced redactable POS blockchain scheme.

the specific number of votes that the member has. As a result, the leader has access to the voting information of all previous CMs for each slot. Based on the historical CMs' voting information obtained, the leader can statistically determine the historical probability distribution of members' types. Assuming that the stakes of blockchain users do not change over time, leader can infer the current probability distribution based on the historical distribution. The leader's goal is to maximise profits through the validating and voting process as follows:

$$\max_{(R_g,F_g)} U_l = \sum_{g=1}^{G} \left(|\mathcal{U}|P_g\right)\left(c^{'}F_g - R_g\right). \tag{3}$$

(2) Utility of CMs: For a CM of type g, based on the signed contract, the utility function is defined as follows:

$$U_m = \theta_g v\left(R_g\right) - cF_g, \tag{4}$$

where $v(R_g)$ is a monotonically increasing valuation function of the incentive R_g for a CM of type g, where $v(0) = 0$, $(\partial v/\partial R) > 0$, and $(\partial^2 v/\partial R^2) < 0$, and c is the unit cost of computing resources spent by a CM. The utility of CMs is the reward received from the leader minus the cost expended. However, CMs wish to maximise their utility by minimising the resource consumption in the validation and voting process. Specifically, the goal of a CM of type g is to maximise his utility, denoted as follows:

$$\max_{(R_g,F_g)} U_m = \theta_g v\left(R_g\right) - cF_g, \forall g \in \{1,\ldots,G\}. \tag{5}$$

(3) Contract Feasibility: Given that the utility function for a certain type CM is defined as equation (4), the contract theory suggests that each contract item for CMs must satisfy the following principles of individual reasonableness (IR) and incentive compatibility (IC) in order for a contract to be feasible. IR implies that a CM will join the block verification and voting when he receives a non-negative utility, i.e.,

$$\theta_g v\left(R_g\right) - cF_g \geq 0, \forall g \in \{1,\cdots,G\}. \tag{6}$$

IC is when a CM of type g, to maximise utility, will only choose the contract (R_g, F_g) over all other contracts (R_{g^-}, F_{g^-}).

$$\theta_g v\left(R_g\right) - cF_g \geq \theta_g v\left(R_{g^-}\right) - cF_{g^-}, \forall g, g^- \in \{1,\cdots,G\}, g \neq g^-. \tag{7}$$

Furthermore, all the rewards that a leader can offer will not exceed $R_{\max}$ which is the transaction fee given by blockchain users for editable transactions on the chain. Thus, we have [18]

$$\sum_{g=1}^{G} |\mathcal{U}|P_g R_g \leq R_{\max}, \forall g \in \{1,\cdots,G\}. \tag{8}$$

According to the constraints given above, the optimization problem can be expressed as

$$\begin{aligned} &\max_{(R_g,F_g)} U_l = \sum_{g=1}^{G}\left(|\mathcal{U}|P_g\right)\left(c^{'}F_g - R_g\right), \\ &\text{s.t.,} \\ &\theta_g v\left(R_g\right) - cF_g \geq 0, \forall g \in \{1,\ldots,G\}, \\ &\theta_g v\left(R_g\right) - cF_g \geq \theta_g v\left(R_{g^-}\right) - cF_{g^-}, \forall g, g^- \in \{1,\cdots,G\}, g \neq g^-, \\ &\sum_{g=1}^{G} |\mathcal{U}|P_g R_g \leq R_{\max}, \forall g \in \{1,\cdots,G\}. \end{aligned} \tag{9}$$

5.2. Optimal Contract for Handling Information Asymmetry. The problem represented equation (9) is not a convex optimization problem. The main difficulty in solving this problem is how to reduce the number of incentive constraints [19]. As the number of IR and IC constraints is N and $N(N-1)$, respectively, we need to simplify the constraints until they are easy to solve. Similar to [18, 20], the number of constraints can only be effectively reduced if the utility function of the CMs satisfies the Spence–Mirrlees property. Luckily, the designed utility function equation (4) satisfies the condition. Thus, equation (9) can be solved with the following steps:

Lemma 1 (Monotonicity Condition). *The incentive R must be monotonically increasing with respect to the type θ of a CM.*

Proof. According to the IC constraint (7), for CMs of type i and type j, where $\theta_i \neq \theta_j$, we can have

$$\begin{aligned} \theta_i v(R_i) - cF_i &\geq \theta_i v\left(R_j\right) - cF_j, \\ \theta_j v\left(R_j\right) - cF_j &\geq \theta_j v(R_i) - cF_i. \end{aligned} \tag{10}$$

Furthermore, we get

$$\left(\theta_i - \theta_j\right)\left[v(R_i) - v\left(R_j\right)\right] \geq 0. \tag{11}$$

Since $(\partial v/\partial R) \geq 0$, whenever $\theta_i > \theta_j$, there must be $R_i \geq R_j$.

Next, we consider three types, i.e., $\theta_{i-1} \leq \theta_i \leq \theta_{i+1}$, and the following constraints which can be called local downward incentive constraints (LDICs).

$$\theta_{i+1} v\left(R_{i+1}\right) - cF_{i+1} \geq \theta_{i+1} v(R_i) - cF_i, \tag{12}$$

$$\theta_i v(R_i) - cF_i \geq \theta_i v\left(R_{i-1}\right) - cF_{i-1}. \tag{13}$$

The equation (13) together with $R_i \geq R_{i-1}$ implies $\theta_{i+1} v(R_i) - cF_i \geq \theta_{i+1} v(R_{i-1}) - cF_{i-1}$. This in turn implies that for type θ_{i+1}, the downward incentive constraint and contract term (R_{i-1}, F_{i-1}) are also satisfied.

$$\theta_{i+1} v\left(R_{i+1}\right) - cF_{i+1} \geq \theta_{i+1} v\left(R_{i-1}\right) - cF_{i-1}. \tag{14}$$

Thus, we can reduce the set of downward incentive constraints to a set of LDICs and the monotonicity condition $R_i \geq R_{i-1}$. It is easy to show that the above approach also holds for the upward incentive constraint set. □

Lemma 2. *LDICs are tight at the optimum point when the monotonicity condition is satisfied.*

Proof. We start by ignoring the set of local upward incentive constraints and concentrate only on the monotonicity of incentive and the set of LDICs. According to the converse method, if the LDIC for some type θ_i is not tight, we have

$$\theta_i v(R_i) - cF_i > \theta_i v\left(R_{i-1}\right) - cF_{i-1}. \tag{15}$$

In this case, the leader can adjust the contract by raising F_i until $\theta_i v(R_i) - cF_i = \theta_i v(R_{i-1}) - cF_{i-1}$.

Based on the above inferences, we can transform equation (9) into

$$\begin{aligned} &\max_{\left(R_g, F_g\right)} U_l = \sum_{g=1}^{G}\left(|\mathcal{U}|P_g\right)\left(c^{'}F_g - R_g\right), \\ &\text{s.t.,} \\ &\theta_1 v(R_1) - cF_1 = 0, \\ &\theta_g v\left(R_g\right) - cF_g = \theta_g v\left(R_{g-1}\right) - cF_{g-1}, \forall g \in \{2, \cdots, G\}, \\ &0 \leq R_1 \leq \cdots \leq R_g \leq \cdots \leq R_G, \\ &\sum_{g=1}^{G} |\mathcal{U}|P_g R_g \leq R_{\max}, \forall g \in \{1, \cdots, G\}. \end{aligned} \tag{16}$$

The standard procedure for solving the equation (16) is to first solve this optimization problem without the monotonicity constraint and then check that the resulting solution satisfies the monotonicity condition [19]. By iterating over the IC and IR constraints, we can obtain

$$F_g = \frac{\left[\theta_1 v(R_1) + \sum_{i=1}^{g} \Delta_i\right]}{c}, \forall g \in \{1, \cdots, G\}. \tag{17}$$

Let $\Delta_i = \theta_i[v(R_i) - v(R_{i-1})]$, $\forall i \in \{2, \cdots, G\}$, $\Delta_1 = 0$. Therefore, equation (16) can be replaced by

$$\begin{aligned} &\max_{R_g, g \in \{1, \cdots, G\}} \sum_{g=1}^{G-1} \left\{ |\mathcal{U}| \frac{c^{'}}{c} v\left(R_g\right)\left[\theta_g \sum_{i=g}^{G} P_i - \theta_{g+1} \sum_{i=g+1}^{G} P_i\right] - |\mathcal{U}|P_g R_g \right\} + |\mathcal{U}| \frac{c^{'}}{c} P_G \theta_G v\left(R_G\right) - |\mathcal{U}|P_G R_G \\ &\text{s.t.,} \\ &\sum_{g=1}^{G} |\mathcal{U}|P_g R_g \leq R_{\max}, \forall g \in \{1, \cdots, G\}. \end{aligned} \tag{18}$$

To solve equation (18), we let $Z_g = |\mathcal{U}|(c^{'}/c) v(R_g)[\theta_g \sum_{i=g}^{G} P_i - \theta_{g+1} \sum_{i=g+1}^{G} P_i] - |\mathcal{U}|P_g R_g$. For each type $\theta_g, g \in \{1, \cdots, G-1\}$, we find an $\widetilde{R_g}$ to maximise the value of Z_g. While for type θ_G, we maximise $|\mathcal{U}|(c^{'}/c) P_G \theta_G v(R_G) - |\mathcal{U}|P_G R_G$ to find $\widetilde{R_G}$. As mentioned before, $(\partial^2 v/\partial R^2) < 0, Z_g$ is a concave function when $|\mathcal{U}|(c^{'}/c)[\theta_g \sum_{i=g}^{G} P_i - \theta_{g+1} \sum_{i=g+1}^{G} P_i] > 0$. Because the sum of concave functions is concave and the constraint is affine,

equation (18) is a convex optimization problem. We assume that the types of CMs obey a uniform distribution so that monotonicity is satisfied [19, 20]. Otherwise, we use the infeasible subsequence substitution algorithm to find the final tuple $(\tilde{R}, \tilde{F})$ [21]. □

5.3. Security against Sybil Attack. A malicious CM may spawn multiple nodes $N = \{n_1, n_2, \ldots, n_q\}$ under his control to participate in validation and voting. He distributes his own votes to these nodes in order to gain lager utility. We can describe the type of CM and the type of these nodes as θ_{g^*} and $\{\theta_{g_1^*}, \ldots, \theta_{g_q^*}\}$ where $\theta_{g^*} = \theta_{g_1^*} + \ldots + \theta_{g_q^*}$.

Theorem 3. *A contract (R_{g^*}, F_{g^*}) according to equation (5) is resistant to Sybil attack.*

Proof. Substituting the left and right sides of the above inequality back into the committee's utility function equation (4), the collation gives

$$\begin{aligned} UC &= U_m(g^*) - \sum_{k=1}^{q} U_m(g_k^*) \\ &= \theta_{g^*} v_{g^*} - \sum_{k=1}^{q} \left(\theta_{g_k^*} v_{g_k^*}\right) - c\left(F_{g^*} - \sum_{k=1}^{q} F_{g_k^*}\right). \end{aligned} \tag{19}$$

For convenience, we abbreviate $v(R_g)$ as v_g. Furthermore, we take equation (17) into equation (19) to get

$$c\left(F_{g^*} - \sum_{k=1}^{q} F_{g_k^*}\right) = \sum_{i=1}^{g^*} \theta_i (v_i - v_{i-1}) - (q-1)\theta_1 v_1 - \sum_{k=1}^{q}\sum_{i=1}^{k} \theta_i (v_i - v_{i-1}). \tag{20}$$

Since $\sum_{i=1}^{g^*} \theta_i (v_i - v_{i-1}) < \sum_{i=1}^{g^*} \theta_i v_i - \sum_{i=1}^{g^*} \theta_{i-1} v_{i-1}$, we have

$$UC > (q-1)\theta_1 v_1 - \sum_{k=1}^{q} \theta_{g_k^*} v_{g_k^*} + \sum_{k=1}^{q}\sum_{i=1}^{k} \theta_i (v_i - v_{i-1}). \tag{21}$$

Since $\sum_{k=1}^{q}\sum_{i=1}^{k} \theta_i (v_i - v_{i-1}) > 0$, a rational CM's UC must be a positive utility when $(q-1)\theta_1 v_1 - \sum_{k=1}^{q} \theta_{g_k^*} v_{g_k^*} > 0$. As a result, we have designed the contract to be well protected against Sybil attack under certain conditions. □

6. Simulation Results

Firstly, this section evaluates the proposed incentive mechanism based on contract theory through simulation. Then, the characteristics of this paper are compared with those of the schemes mentioned in the paper.

All of our experiment is run on desktop with AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz CPU and 16.0 GB RAM on Windows 10. For comparison purposes, we compare the present incentive mechanism under information asymmetry with another incentive mechanism without information asymmetry, which refers to the leader knowing the specific type of each CM. Obviously, optimal design without information asymmetry is the best result we can achieve.

It is assumed that there are 500 CMs and a leader. The type of each CM follows a uniform distribution. They are classified into 20 different types according to the votes, so the probability of each member being a certain type is 0.05. Parameter $c' = 5$ or $c = 1$. As a contract publisher, a leader generates contract items based on the information that has been obtained and sends them to each CM. Each stakeholder chooses to sign a contract and then acts as voter and verifier to execute the contract. Every CM completes his task honestly will receive a reward from the leader.

6.1. Contract Feasibility. Figures 3(a) and 3(b) show that incentive and resource improve with node type, which reflects the monotonicity of our system. The difference is that the incentive for our contract is a concave function with respect to the node type, whereas the incentive without information asymmetry is a linear function. We can see that under no information asymmetry, when the miner knows the specific type of node, it can obtain higher resources with lower rewards.

The utility of the node with different types ranging from 17 to 19 is presented in Figure 3(c). The results show that utility is only maximised when a node chooses a contract item designed for his type and the utility of node is non-negative. The former accounts for the IC constraint, and the latter verifies the IR constraint.

6.2. Contract Performance. Figure 4 reveals that different types of nodes bring different utilities to miner. The higher the node type, the higher utility it can bring. Figure 4(a) presents that the utility under no information asymmetry is an upper bound on the utility under information symmetry. This is because the miner knows all the information about the node in the former condition. Figure 4(b) displays that the optimization utility of our mechanism is higher than the utility without information asymmetry and that this utility remains zero. The reason for this has already been explained in the previous chart. Thus, the incentive of information asymmetry protects nodes from being over utilized. Figures 4(a) and 4(c) show the same performance that is because the utility is still highest with no information asymmetry, but we strive for some reward for the nodes.

6.3. Comparison of Solutions. In order to better reflect the innovation and necessity of the enhanced redactable POS blockchain solution proposed in this paper, we compare this

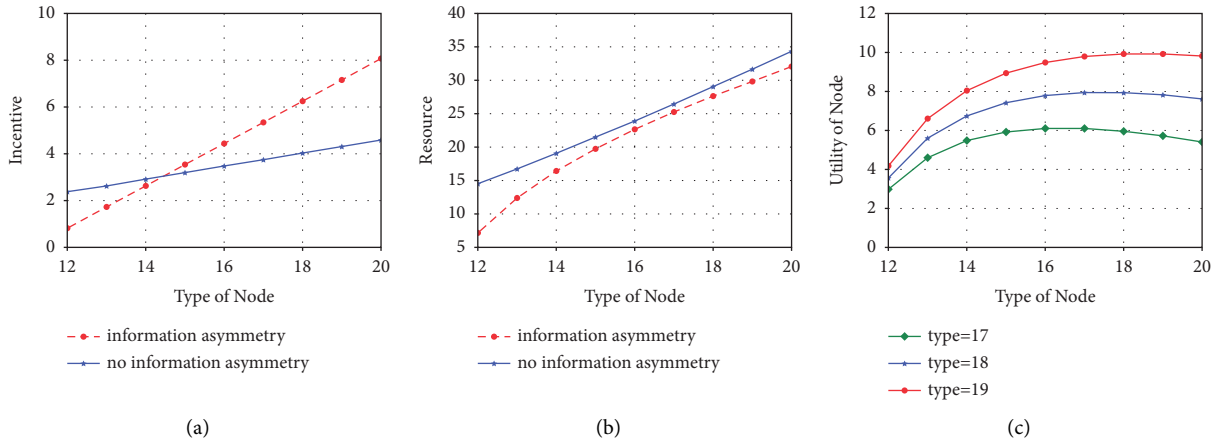


Figure 3: Contract feasibility: (a) incentive, (b) reward, and (c) utility of node.

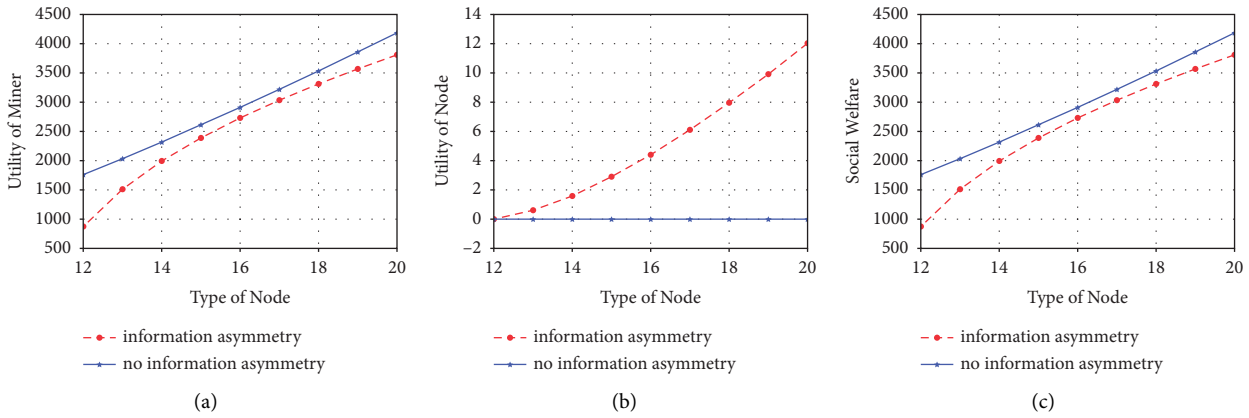


Figure 4: Contract performance of different type node: (a) utility of miner, (b) utility of node, and (c) social welfare.

Table 2: Comparison of editable blockchain solutions.

Features	Works					Ours
	Ateniese et al. EuroS&P′17 [4]	Derler et al. NDSS'19 [11]	Deuber et al. S&P′19 [16]	Xu et al. TIFS'21 [15]	Li et al. TDSC'22 [5]	
Decentralization	✗	✗	✓	✗	✓	✓
Without MPC	✗	✗	✓	✓	✓	✓
Public verifiability	✗	✗	✓	✓	✓	✓
Incentive mechanism	✗	✗	✗	✓	✗	✓

Scheme characteristics: ✓ means fully realized, ✗ means not realized.

paper with the existing research works in terms of the following four features: decentralization, without MPC, public verifiability, and incentive mechanism. The results are shown in Table 2.

As can be seen from the above table, this paper makes an innovative design to add incentives based on the literature [5]. In terms of decentralization and without MPC, existing works [4, 11] and [15] require a central entity for the issuance of editing rights, and some of them also require MPC for trapdoor management, while this paper is a decentralized scheme based on voting to reach consensus. In terms of public verifiability, the disclosure or nondisclosure of the chameleon hash trapdoor determines whether the edited block satisfies public verifiability. Overall, a voting-based editable blockchain solution can achieve decentralization, without MPC, public verifiability, and this paper adds an incentive mechanism to effectively engage enough CMs to honestly participate in verifying and voting on editable blocks. As a result, the research work in this paper further improves the security of the block editing process compared to existing studies.

7. Conclusions

This paper proposes a contract theory-based incentive mechanism on voting-based redactable POS blockchains to deal with the issue of insufficient committee incentives. To

demonstrate the effectiveness of the mechanism, we compare the feasibility and performance with a contract design that does not consider information asymmetry. The experimental results show that the incentive mechanism can effectively attract enough high-stakes CMs to honestly join the validation and voting of editable blocks. In addition, the mechanism can defend against Sybil attack. Therefore, the present incentive for CMs with high stake in the voting-based editing POS blockchain solution is practical, as it allows these members to receive the rewards they deserve. In the future, we will investigate incentive mechanisms with broader incentive coverage so that CMs with lower voting weights can also receive the rewards they deserve.

Data Availability

The related data used to support the findings of this study have been deposited in the incentive-mechanism repository (https://github.com/hippo212/incentive-mechanism).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was in part supported by the Guangdong Key R&D Plan 2020 (Grant no. 2020B0101090002), National Natural Science Foundation of China (Grant no. 61932011), Guangdong Basic and Applied Basic Research Foundation (Grant no. 2019B1515120010), Guangdong Key Laboratory of Data Security and Privacy Preserving (Grant no. 2017B030301004), National Key R&D Plan 2020 (Grant no. 2020YFB1005600), National Joint Engineering Research Center of Network Security Detection and Protection Technology (Grant no. 2016B010124009), and Guangdong Provincial Special Funds for Applied Technology Research and Development and Transformation of Important Scientific and Technological Achieve (Grant no. 2017B010124002).

References

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Decentralized Business Review*, Article ID 21260, 2008.

[2] J. M. L. Alfonsín, "Argentina: the right to be forgotten," *The Right to Be Forgotten*, pp. 239–248, Springer, Berlin, Germany, 2020.

[3] P. Voigt and A. Von dem Bussche, "The Eu General Data protection Regulation," *A practical guide*, Springer International Publishing, vol. 103152676, pp. 10–5555, New York, NY, USA, 1 edition, 2017.

[4] G. Ateniese, B. Magri, D. Venturi, and E. Andrade, "Redactable blockchain–or–rewriting history in bitcoin and friends," in *Proceedings of the 2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 111–126, IEEE, Paris, France, April 2017.

[5] X. Li, J. Xu, L Y. Yin et al., "Escaping from Escaping From Consensus: Instantly Redactable Blockchain Protocols in Permissionless Settingonsensus: Instantly Redactable Blockchain Protocols in Permissionless setting," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–20, 2022.

[6] S. Micali, M. Rabin, and S. Vadhan, "Verifiable random functions," in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pp. 120–130, IEEE, New York, NY, USA, October 1999.

[7] G. Rathee, A. Sharma, R. Iqbal, M. Aloqaily, N. Jaglan, and R. Kumar, "A blockchain framework for securing connected and autonomous vehicles," *Sensors*, vol. 19, no. 14, p. 3165, 2019.

[8] G. Rathee, F. Ahmad, N. Jaglan, and C. Konstantinou, "A secure and trusted mechanism for industrial IoT network using blockchain," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1894–1902, 2023.

[9] R. Krishnamurthy, G. Rathee, and N. Jaglan, "An enhanced security mechanism through blockchain for E-polling/counting process using IoT devices," *Wireless Networks*, vol. 26, no. 4, pp. 2391–2402, 2020.

[10] Y. Cai, G. Fragkos, E. E. Tsiropoulou, and A. Veneris, "A truth-inducing sybil resistant decentralized blockchain oracle," *IEEE*, in *Proceedings of the 2020 2nd conference on blockchain research & applications for innovative networks and services (BRAINS)*, pp. 128–135, Paris, France, September 2020.

[11] D. Derler, S. Kai, and D. Slamanig, "Fine-grained and controlled rewriting in blockchains: chameleon-hashing gone attribute-based," in *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*, Diego, CA, USA, February 2019.

[12] Y. Tian, N. Li, Y. Li, P. Szalachowski, and J. Zhou, "Policy-based chameleon hash for blockchain rewriting with black-box accountability," in *Proceedings of the Annual Computer Security Applications Conference*, pp. 813–828, Austin, TX, USA, December 2020.

[13] G. Panwar, R. Vishwanathan, and S. Misra, "Retrace: revocable and traceable blockchain rewrites using attribute-based cryptosystems," in *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, pp. 103–114, Spain, June 2021.

[14] Y. Jia, S.-F. Sun, Y. Zhang, Z. Liu, and D. Gu, "Redactable blockchain supporting supervision and self-management," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pp. 844–858, Hong Kong, May 2021.

[15] S. Xu, J. Ning, J. Ma, X. Huang, and R. H. Deng, "K-time modifiable and epoch-based redactable blockchain," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4507–4520, 2021.

[16] D. Deuber, B. Magri, and S. A. K. Thyagarajan, "Redactable blockchain in the permissionless setting," in *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, pp. 124–138, IEEE, San Francisco, CA, USA, May 2019.

[17] J. Kang, Z. Xiong, D. Niyato, P. Wang, D. Ye, and D. I. Kim, "Incentivizing consensus propagation in proof-of-stake based consortium blockchain networks," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 157–160, 2019.

[18] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles:

optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[19] P. Bolton and M. Dewatripont, *Contract Theory*, MIT press, Cambridge, MA, USA, 2004.

[20] Y. Zhang, L. Song, W. Saad, Z. Dawy, and Z. Han, "Contract-based incentive mechanisms for device-to-device communications in cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2144–2155, 2015.

[21] L. Gao, X. Wang, Y. Xu, and Q. Zhang, "Spectrum trading in cognitive radio networks: a contract-theoretic modeling approach," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 843–855, 2011.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 4941214, 13 pages
https://doi.org/10.1155/2022/4941214

Research Article

A Blockchain-Based Personal Health Record System for Emergency Situation

Yuan Liu,[1] **Yan Du**,[2,3] **Yanan Zhang**,[2,3] **Yuan Li**,[4] **Leung Cyril**,[5] **Chunyan Miao**,[6] **Qingfeng Tan**,[1] **and Zhihong Tian**[1]

[1]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, Guangdong 510006, China*
[2]*Software College, Northeastern University, Shenyang, Liaoning 110169, China*
[3]*China-Singapore International Joint Research Institute, Guangzhou, Guangdong 510663, China*
[4]*Qi Lu Hospital of Shandong University, Jinan, Shandong 250012, China*
[5]*Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, Nanyang Technological University, Singapore*
[6]*School of Computer Science and Engineering, Nanyang Technological University, Singapore*

Correspondence should be addressed to Yuan Li; liyuan91700@163.com and Qingfeng Tan; tqf528@gzhu.edu.cn

Received 23 July 2022; Accepted 20 September 2022; Published 8 October 2022

Academic Editor: Yinbin Miao

A personal health record (PHR) system stores personal health-related information, which can assist physicians in quickly forming appropriate treatment plans in emergency situations. Because a PHR contains lots of sensitive information, the patients are only willing to share their records with authorized doctors with their permission. There are three main challenging issues: (1) it is costly to store and maintain the growing PHRs data; (2) the existing PHR systems still face the privacy leakage risk during data transmission and access control processes; and (3) the response speed cannot meet the need in an emergency situation, especially when the patients are unconscious. In this paper, based on the permissioned blockchain Hyperledger Fabric, we propose a PHR management system preserving patients' privacy and also supporting emergency access. In the system, we use reencryption technology and the anonymous identity mapping mechanism to protect patient privacy and use smart contracts to define access control strategies in an emergency situation. Furthermore, we use a quick response code and bloom filter to optimize system performance. The security analysis and experimental results show that our proposed framework guarantees the privacy of patient data from multiple aspects while improving the efficiency with which doctors can obtain PHR information.

1. Introduction

A personal health record (PHR) including healthcare history, medical records, allergy history, and genetic diseases is an important property of each patient [1]. Patients own their PHRs, and they do not have to disclose all their records to doctors whenever they seek medical treatment, unless in the case of major diseases. Meanwhile, doctors should also be allowed to check their patients' PHRs, even in emergency situations, when the patients may be unconscious and their lives are in danger [2].

Many systems have been designed to manage and access patients' personal health records based on traditional centralized databases [3, 4] or blockchain [5]. When a patient seeks treatment from a doctor, the doctor can view the patient's personal health information under the patient's permission. However, these systems still have several challenging problems, especially in an emergency situation. Firstly, its storage is costly for a healthcare service institute or hospital in recording and maintaining the fast-growing volume of personal information. In order to overcome the storage cost, the health recording service may only promise to maintain the records happening in the recent 3 years or 5 years and discard the older records. Secondly, the health records are stored in a centralized server and the data is at risk to be leaked when the data is transmitted or accessed by doctors. Since health records contain highly private and sensitive information, data leakage can result in exposure to

privacy. Thirdly, the traditional access control methods bear relatively long time delay without considering the emergency requirements; meanwhile, the quick access mechanism is credential for patients.

Aiming at solving the above three problems, this paper designs and implements a personal health record system based on blockchain. The system is composed of three main modules: personal health records storage, emergency access management, and encrypted transmission. The increasing data of all patients are stored in the distributed file system IPFS [6, 7] and, to avoid a vast amount of data occupying massive blockchain space, only the hash of the uploaded personal health record file on IPFS is stored in the blockchain. In addition, the access control strategy is set on the blockchain and executed by smart contracts automatically. While the unconscious patient with certain identifying information is under treatment, smart contracts check the identity of the emergency doctor who applies for the PHR file according to the access control list (ACL). ACL determines whether the emergency physician is allowed to access the data by screening the doctor's ID. An application is launched after obtaining the information about the patient (the information is not the patient's personal health record information, but the information used for identity authentication, while letting the system know which patient's information to obtain). This information is described as a quick response code in this paper.

The most important part of this system is the protection of patients' personal health records. The protection consists of three parts: anonymous storage of encrypted private keys, the encryption of personal health record files, and the encryption of message transmission. The anonymous identity mapping mechanism [8] makes it impossible for anyone other than the patient to know the encrypted file corresponding to the private key, so, even if someone gets the private key, they do not know which file to decrypt. This part ensures the security of the smart contract execution that decrypts the private key. The second part ensures the privacy and security of the personal health record file. The personal health record file is encrypted by the key from AES algorithm [9] and it is encrypted again by the public key of patient (notice that the key from the AES algorithm is different from the public key and private key of patient). Encrypted files are stored on IPFS that returns hash values to the patient. Furthermore, the patient's private key is encrypted and then stored on the blockchain (the private key is encrypted and anonymized to prevent leakage, and it is decrypted during the execution of the smart contract). The third part guarantees the security of message transmission during the application of data. When an authorized doctor applies the patient's data, the smart contract will send the corresponding hash value of the patient's encrypted personal health record file. The doctor can ask for the file from IPFS through the given hash. After obtaining the double-encryption file, the doctor will apply for the decryption of the patient's private key to completely decrypt the original PHR file encrypted from the AES algorithm and the patient's public key. What is more, the smart contract is launched to reencrypt the decrypted AES key by the doctor's public key and return it to the doctor. The doctor can then decrypt it with his/her private key and see the patient's original PHR file.

Meanwhile, the design of this system also considers the response time to race against time for the patient's life. Each patient can generate a quick response code that includes his/her basic information, representing his/her identity. Doctors enter the system by scanning codes to save the time of finding and inputting patient information. Furthermore, the bloom filter [10] is used to store the access control list, which can quickly filter the medical doctor's identification to accelerate the authentication of the doctor's identity.

Our system is a Hyperledger Fabric [11] based blockchain application that realizes data sharing with emergency doctors while ensuring the privacy of patients' personal health records in emergency situations. In addition, according to experimental verification, the performance of our proposed system is greatly optimized, which can effectively accelerate the speed of doctors obtaining a patient's PHR. The contributions of this paper are as follows:

(i) We achieve doctors' access without the authorization of the patient's supervision in an emergency situation through escrowing the patient's encrypted private key on the blockchain and setting the access control list on the smart contract.

(ii) We use symmetric encryption, asymmetric encryption, and reencryption technologies to protect the privacy of patients' PHR and introduce the anonymous identity mapping mechanism to protect the encrypted private key escrowed on the blockchain.

(iii) In order to optimize the system, we introduce the quick response code and bloom filter to accelerate emergency physicians to obtain the patient's PHR data and also design a smart contract to verify that PHR data has only been added but not changed when updating.

The remainder of this article is organized as follows. Section 2 introduces some related work. Section 3 describes the proposed model. Section 4 analyzes the security and privacy of the proposed model. Section 5 presents the results of the simulation experiments and evaluates the performance of the proposed model. Finally, the paper is summarized in Section 6.

2. Related Work

In this section, we summarize some outstanding research work currently solving problems in the PHR system, while discussing the weakness of existing solutions.

In order to better store and share the patient's personal health records, researchers introduced the semitrusted server to implement data storage and proxy reencryption. Li et al. [17] proposed a framework for access control to PHR stored in a semitrusted server, using attribute-based encryption (ABE) technology to encrypt the PHR file of each patient. Users in the PHR system are divided into multiple

security domains, which reduces the complexity of key management for users and ensures a high degree of privacy for patients. Bhatia et al. [18] proposed a lightweight certificateless proxy reencryption scheme to make the PHR system capable of low-power mobile devices, which uses the semitrusted proxy server to perform the reencryption process. Reference [14] proposed a revocable and unpaired ciphertext policy attribute-based encryption for the management of personal health records and added a proxy decryption server to decrypt the partial ciphertext at the decryption end, which can effectively reduce the computational overhead of the decryption end. Although the above models protect the privacy of the PHR system and consider efficiency issues, their designs all rely on the semitrusted server. The server is extremely vulnerable to single-point attacks, which cannot guarantee the security of data.

After discovering the shortcomings of the centralized PHR system, a lot of research work introduced blockchain. Hussien et al. [16] proposed an attribute searchable encryption method based on the smart contract to achieve secure and fine-grained access control. They introduced distributed storage IPFS to avoid storing large amounts of data on the blockchain and used one-to-many encryption to prevent unauthorized users from accessing data stored in IPFS. Wang et al. [19] proposed a new data integrity verifiable PHR sharing scheme based on blockchain. The new scheme uses searchable symmetric encryption and attribute-based encryption technology to achieve fine-grained access control without the involvement of a third party. Thwin and Vasupongayya [20] proposed a blockchain-based PHR model, which uses proxy reencryption and access control list to achieve flexible access control to the patient's PHR, revokes the doctor's access authority by updating the access control list, and uses blockchain to record access logs to ensure data auditability and nontampering. Although the above PHR systems use blockchain technology, they are designed only for conscious patients, and the patient can authorize when the doctor requests access to the data. The emergency situation is not considered, such as how to handle data requests from doctors while the patient is unconscious.

In dealing with emergency access, Huda et al. [12] introduced a new privacy-aware protocol for handling healthcare professionals' access to patient-controlled PHR in emergency situations. It uses an IC card embedded with a patient's emergency access report for strong authentication. The method of storing data on the IC card is risky. If the IC card is lost, it will cause irreparable losses. The emergency access policy designed in [15] is to preset a timeout period. If the patient rejects the emergency request, the patient is conscious and the doctor needs to be authorized to access normally. If the patient does not operate over time, the system will approve an urgent request. The setting of the timeout period here is a problem. Reference [21] proposed three verification methods for the PHR system in emergency situations. The first is that the telecommunication provider determines whether the patient's telecommunication equipment is within a reasonable distance of the hospital location. If reasonable, the telecommunication provider provides the patient's key sharing. The second is to send a shared key request to emergency contacts if the patient does not respond within a reasonable time. The third is to combine the first and second methods; if the patient's emergency contacts do not answer the phone, the first method will be used. References [22, 23] and [24] used the second method but cannot guarantee that emergency contacts respond to data requests rapidly. Reference [25] proposed a context-aware technology to realize automatic transmission of authorization in the PHR system. According to different roles, the permissions granted are different. Some role owners, such as emergency physicians, obtain additional permissions for certain data objects, thus ensuring access to PHR data when the patient is unconscious. Reference [13] also used a role-based authorization method. Although the duration of emergency doctor access is limited, it does not achieve complete privacy protection, because the patient's PHR information is not private to the emergency doctor regardless of whether the patient is in an emergency. Through the analysis, it can be seen that current access control strategies in the PHR systems under emergency situations have not achieved the desired effect.

Compare our system with the system proposed in the existing papers based on the above analysis from five aspects: decentralized, patient data privacy, access control, considering emergency, and accelerating response. The comparison results are shown in Table 1. It can be seen from the comparison results that part of the systems proposed in the existing papers does not consider solving the problems that exist in emergency situations and, currently, there are few considerations for optimizing the PHR system and accelerating the response speed of the system.

3. The Proposed System Model

In this section, we will introduce our system model in detail, including system architecture, system workflow, and system optimization.

3.1. System Architecture. The architecture of the system model proposed in this paper is shown in Figure 1. Firstly, the patient provides his/her basic information and personal health records. The complete PHR file is encrypted by symmetric encryption algorithm AES and asymmetric encryption algorithm RSA. Then, the encrypted data is uploaded to the distributed storage network IPFS, and the hash value returned by IPFS is recorded on the blockchain. The encrypted private key of the patient is also stored on the blockchain, because of considering the patient's unconscious situation. Here, in order to ensure the security of the encrypted private key, this paper introduces the anonymous identity mapping mechanism to store the encrypted private key on the blockchain. In addition, each patient generates a quick response code (QR) based on his/her basic information to indicate his/her identity. The doctor scans the QR code to get the patients' basic information and quickly enters the system. Smart contracts deployed on the blockchain authenticate the identity of the doctor who enters. The authorized doctor initially obtains the encrypted data. This

Table 1: The comparsion of different PHR system.

PHR system name	Decentralized	Patient data privacy	Access control	Considering emergency	Accelerating response
Huda et al. [12]	✗	✓	✓	✓	✗
Rajupt et al. [13]	✓	✗	✓	✓	✗
Liu and Xu [14]	✗	✓	✓	✗	✗
Son et al. [15]	✓	✓	✓	✓	✗
Hussien et al. [16]	✓	✓	✓	✗	✗
Our system	✓	✓	✓	✓	✓

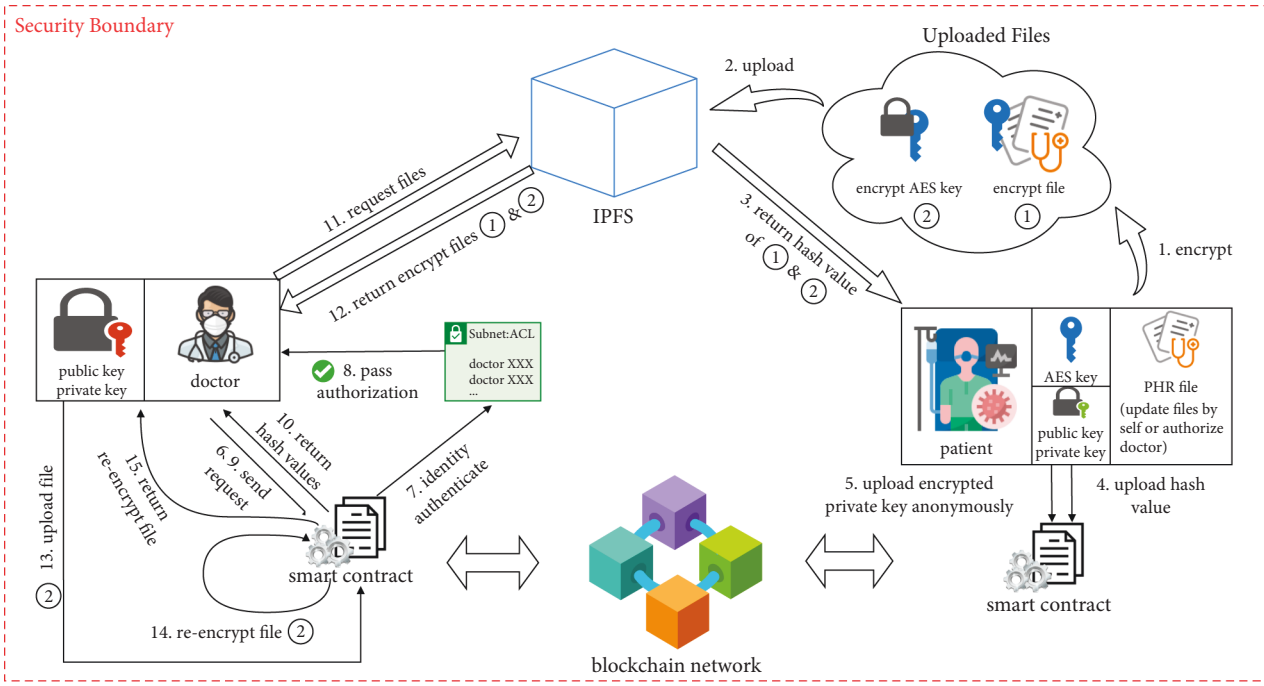


Figure 1: The proposed system architecture.

data is first reencrypted by the smart contract and then decrypted by the doctor. For the update of the PHR data, the patient can update the data himself/herself, and the doctor can also help the patient update the medical record with the patient's permission. In the update, a smart contract is designed to verify that the data has only been added and not changed. Our system contains the following five entities:

(1) patient is an entity that owns PHR data and wishes to share his/her PHR data with the doctor when necessary. The patient has absolute control over PHR data; he can define access control strategies and decide who can access his/her data. Considering that data must be accessed in an emergency situation, the patient escrows the encrypted private key on the blockchain and provides the corresponding decryption method in the smart contract. Besides, every patient has a quick response code that represents his/her identity.

(2) doctor is an entity that applies for access to PHR data and accurately treats patients based on existing data. The doctor quickly enters the system by scanning the QR code to authenticate the identity and then queries the patient's hash values stored on the blockchain based on the information in the QR code. The doctor downloads the corresponding encrypted file on IPFS according to the hash values and then sends the file to the smart contract for reencryption. The doctor finally decrypts the reencrypted file with his/her own private key to get the complete PHR data.

(3) IPFS is an entity used to store encrypted files. It is a distributed file system based on content addressing. After patients upload encrypted files, they will get the corresponding hash values, and the hash value of each file is unique.

(4) Blockchain network is responsible for storing some hash values and recording access logs. The blockchain in this system refers to Hyperledger Fabric, which only allows partial organizations to access it.

(5) Smart contract is an agreement that can automatically perform some functions without the intervention of a third party. The smart contract automatically performs operations such as decryption of the patient's private key, re-encryption of PHR data, and verification of data only being added without change.

Table 2: Explanation of notations in different transaction processes.

Notation	Description
PID	The ID represents patient's identity
DID	The ID represents doctor's identity
$\text{AnonymityID}_{\text{pid}}$	The patient's anonymous identity
HNumber	The number of doctors' hospitals
pk_{pid}, sk_{pid}	The patient's public key and private key
pk_{did}, sk_{did}	The doctor's public key and private key
Num	Length of the generated symmetric key
K	AES symmetric key
C	Original PHR file
C'	Updated PHR file
C_1	File with symmetric encrypted PHR
C_1'	Updated symmetric encrypted PHR file
C_2'	File with encrypted K
C_3'	File with reencrypted EnK
$\text{En}sk_{\text{pid}}$	The patient's cyclically encrypted private key
EnK	K's ciphertext after being encrypted by PK_A
EnK′	EnK's reencrypted key
Hash_1	Hash value returned by IPFS uploading file C_1
Hash_2	Hash value returned by IPFS uploading file C_2
Hash_3	Hash value returned by IPFS uploading file C_1'

3.2. System Workflow. In this part, we will introduce the main workflow of the system from the perspective of the patient. First, the patient stores the PHR data. Then, the doctor requests to access the patient's PHR data. Finally, the patient or doctor updates the PHR data. The specific operations in the above three different workflows are mainly introduced in detail: the encryption and uploading operations performed in the storing PHR data process; the downloading, re-encryption, and decryption operations performed in the requesting PHR data process; the authorization setting and the verification of only being added without change operations performed in the process of updating PHR data process. The main notations used in this section are shown in Table 2.

3.2.1. Storing PHR Data. The interaction process of storing PHR data is shown in Figure 2. The main execution process is as follows:

(1) KeyGen1 (PID) $\longrightarrow$ (pk_{pid}, sk_{pid}): the KeyGen1 algorithm is used to delegate a trusted party to generate a pair of keys. It takes as input the patient's identity PID and outputs a key pair pk_{pid}, sk_{pid}.

(2) KeyGen2 (Num) $\longrightarrow$ K: the KeyGen2 algorithm is used to delegate a trusted party to generate a symmetric key to encrypt the PHR file. It takes as input the specified length of the key Num $\in$ {128, 192, 256} and outputs symmetric key K.

(3) Enc1 (K, C) $\longrightarrow$ C_1: the Enc1 algorithm uses K to encrypt the patient's uploaded file based on the symmetric encryption algorithm AES. It takes as input the symmetric key K and the original PHR file C and outputs the encrypted file C_1.

(4) Enc2 (pk_{pid}, K) $\longrightarrow$ C_2: the Enc2 algorithm uses pk_{pid} to encrypt the symmetric key and writes the encrypted key to the file. It takes as input the patient's public key pk_{pid} and symmetric key K and outputs the encrypted file C_2.

(5) Enc3 (sk_{pid}) $\longrightarrow$ $\text{En}sk_{\text{pid}}$: the Enc3 algorithm is run by those who need to share their PHR data based on the cyclic shift encryption algorithm. It takes as input the patient's private key sk_{pid} and outputs the encrypted key $\text{En}sk_{\text{pid}}$.

(6) UploadToIPFS (C_1, C_2) $\longrightarrow$ (hash_1, hash_2): the UploadToIPFS algorithm is run by patients to upload encrypted files to the distributed storage system IPFS. It takes as input encrypted file C_1 and C_2 and outputs two hash values hash_1 and hash_2 which are returned by IPFS.

(7) IdentityMap (PID) $\longrightarrow$ $\text{AnonymityID}_{\text{pid}}$: the IdentityMap algorithm anonymizes the patient's identity, which is only visible to developers after encapsulation. It takes as input the patient's identity PID and outputs an anonymous identity $\text{AnonymityID}_{\text{pid}}$.

(8) InitializeP (PID, hash_1, hash_2, $\text{AnonymityID}_{\text{pid}}$, $\text{En}sk_{\text{pid}}$): the InitializeP algorithm is run by smart contract and stores this information on the blockchain in the form of key-value pairs like {PID, {hash_1, hash_2}} and {$\text{AnonymityID}_{\text{pid}}$, $\text{En}sk_{\text{pid}}$}. The encrypted private key is stored anonymously to prevent the private key from being leaked.

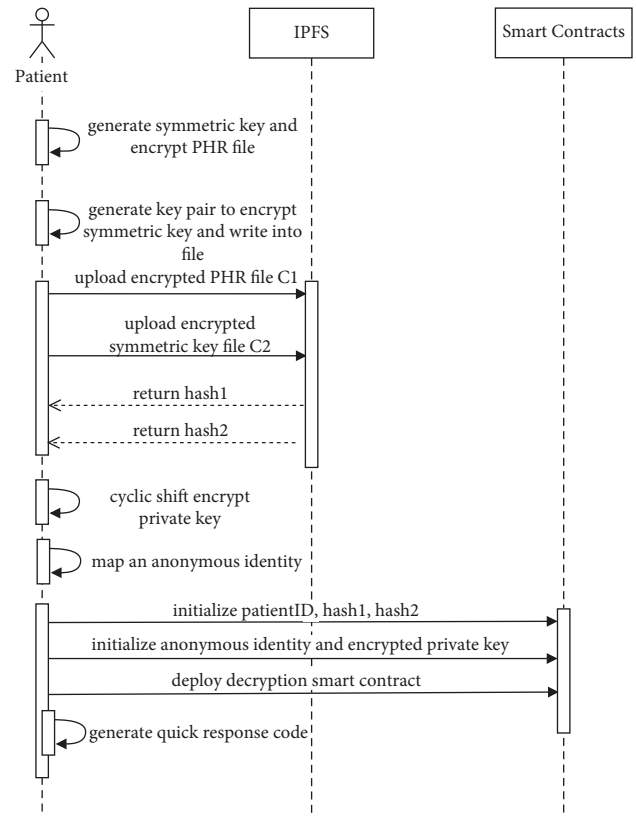


Figure 2: Sequence diagram of storing a PHR.

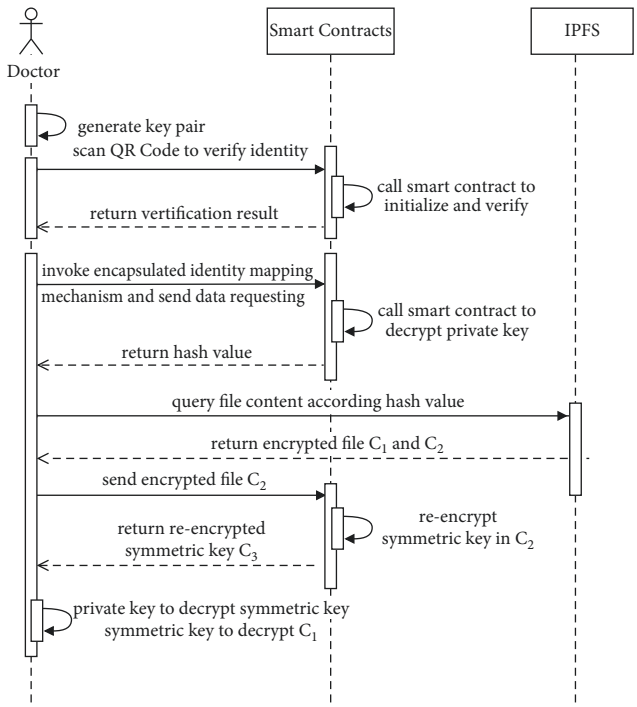


Figure 3: Sequence diagram of requesting a PHR.

After uploading the PHR data, the patient needs to deploy a smart contract for decrypting Ensk_{pid} on the blockchain. The smart contract can be invoked to obtain the decrypted private key. Since the encrypted private key is stored anonymously, the attacker needs a lot of computing power to map the private key to the corresponding user. Meanwhile, the blockchain records who invokes the decryption smart contract. If there is an illegal operation, the patient can pursue the legal responsibility of the attacker after regaining consciousness. At last, the patient can generate a QR code representing his/her identity according to the input PID and the format of the uploaded file, which is convenient for doctors to enter the system quickly.

3.2.2. Requesting PHR Data. The interaction process of requesting PHR data is shown in Figure 3. The doctor enters the system to authenticate his/her identity by scanning the QR code. The QR code can provide the patient's basic information like PID. The main execution process is as follows:

(1) KeyGen3 (DID) $\longrightarrow$ (pk_{pid}, sk_{pid}): the KeyGen3 algorithm is used to delegate a trusted party to generate a pair of keys. It takes as input the doctor's identity DID and outputs a key pair pk_{pid}, sk_{pid}.

(2) InitializeD (DID, HNumber): the InitializeD algorithm is run by a smart contract and stores doctor's information on the blockchain in the form of key-value pairs like {DID, Hnumber}.

(3) Authentication (DID) $\longrightarrow$ True/False: the authentication algorithm is run by a smart contract and used to authenticate the doctor's identity. Firstly, the algorithm judges whether the doctor's ID contains "ED," "ed," "Ed," or "eD" and then checks whether the qualified doctor's ID is in the access control list. This algorithm takes as input the doctor's identity DID and outputs the authentication result true or false.

(4) QueryInfo (PID) $\longrightarrow$ ($hash_1$, $hash_2$): the QueryInfo algorithm is to query the corresponding information on the blockchain. It takes as input the patient's identity PID and outputs hash values of encrypted files $hash_1$ and $hash_2$.

(5) DownloadFile ($hash_1$, $hash_2$) $\longrightarrow$ (C_1, C_2): the DownloadFile algorithm is to download the patient's encrypted files from IPFS by hash values. It takes as input files' hash values $hash_1$ and $hash_2$ and outputs encrypted files C_1 and C_2.

(6) Dec1 (PID) $\longrightarrow$ sk_{pid}: the Dec1 algorithm maps the patient's anonymous identity according to the encapsulated identity mapping algorithm IdentityMap (PID) and then queries the patient's encrypted private key Ensk_{pid} on the blockchain according to this anonymous identity. The algorithm is run by a smart contract to automatically decrypt the patient's encrypted private key. This algorithm takes as input the patient's identity PID and outputs the patient's private key sk_{pid}.

(7) ReEnc (C_2, pk_{pid}, sk_{pid}) $\longrightarrow$ C_3: the ReEnc algorithm uses pk_{pid} and sk_{pid} to generate new key to reencrypt file C_2. This algorithm is run by a smart contract. It takes as input file C_2, doctor's public key pk_{pid}, and patient's private key sk_{pid} and outputs reencrypted file C_3.

(8) Dec2 (C_1, C_3, sk_{pid}) $\longrightarrow$ C: the Dec2 algorithm is used to decrypt the file to obtain the complete PHR file. This algorithm uses sk_{pid} to decrypt file C_3 to obtain K and then uses K to decrypt file C_1 to obtain the original PHR file C.

3.2.3. Updating PHR Data. The interaction process of updating PHR data is shown in Figure 4. Patients can update the PHR data themselves or they can authorize physicians to update through settings. The following main execution process is introduced using patient updating as an example.

(1) AuthorizeSet (PID) $\longrightarrow$ True/False: the AuthorizeSet algorithm is for the conscious patient to set the doctor's permission of updating his/her PHR data. This algorithm can reset the authorization setting when necessary for the patient. It takes as input the patient's identity PID and outputs authorization setting result as true or false.

(2) Enc1 (K, C') $\longrightarrow$ C_1': the Enc1 algorithm is the same as the algorithm in the third step of the process of storing PHR data; only the encrypted data is different. It takes as input the symmetric key K and updated PHR file C' and outputs the encrypted file C_1'.

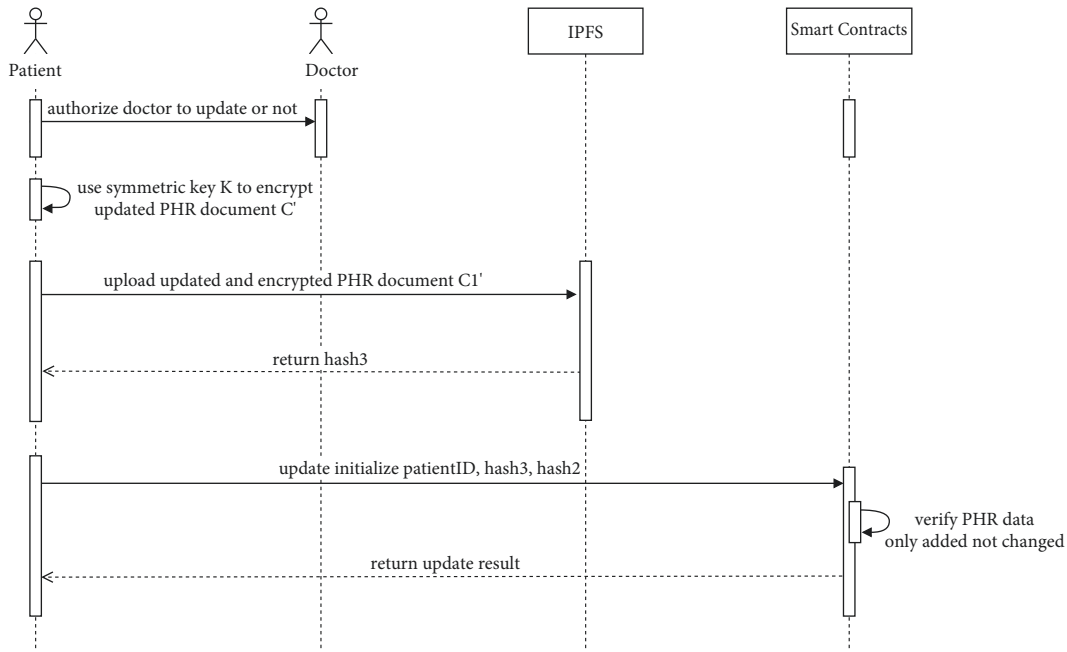


Figure 4: Sequence diagram of updating a PHR.

(3) UploadToIPFS (C_1') $\longrightarrow$ hash_3: the UploadToIPFS algorithm is the same as the algorithm in the fourth step of the process of storing PHR data; only the uploaded file content is different. It takes as input the encrypted file C_1' and outputs the hash value hash_3 returned by IPFS.

(4) VerifyOnlyAdded (C_1', C_1) $\longrightarrow$ True/False: the VerifyOnlyAdded algorithm is run by a smart contract and verifies the updated PHR file before updating data on the blockchain. This algorithm is to compare the contents of two encrypted files to determine whether the PHR file has only been added but not changed. It takes as input encrypted files C_1' and C_1 and outputs the validation results as true or false.

(5) InitializeP (PID, hash_3, hash_2, $\text{AnonymityID}_{\text{pid}}$, Ensk_{pid}): the InitializeP algorithm can only run if the returned result of the VerifyOnlyAdded (C_1', C_1) algorithm is true. This initialization algorithm just updates the hash value of the saved PHR file, and the rest of the parameters remain unchanged. The data stored in key-value pairs on the blockchain is modified to {PID, {hash_3, hash_2}} and {$\text{AnonymityID}_{\text{pid}}$, Ensk_{pid}}.

For doctors, they enter the PHR data acquisition system after scanning the patient's QR code. If the current patient has authorized the doctor to update, the doctor can click the updating entry to update the data according to the above process. Otherwise, there is no update entry in the PHR data acquisition system.

3.3. System Optimization. In order to ensure that the response speed of the system meets the need in emergency situations, our system has made the following two optimizations.

Figure 5: An example of the QR code.

3.3.1. Quick Response Code. The patient generates a QR code according to his/her ID and the format of the uploaded PHR file. The QR code must include the PHR file format because only the content of the file can be queried from IPFS based on the corresponding hash value. If the content is not saved in the correct formatted file, the content will be garbled. Figure 5 is an example of the QR code. QR code does not store private information, so there is no issue of privacy leakage. The patient can print out his/her QR code and carry it with him/her. In case of an emergency, legal doctors can directly scan the code to enter the doctor authentication interface, saving time for the doctor to find and input the patient's identity information. Furthermore, this QR code can provide effective identity information for the patient when the patient does not bring the identity card or medical card, which brings convenience to the medical treatment.

3.3.2. Bloom Filter. When a doctor applies for access to the patient's PHR, he first needs to authenticate the identity. 100,000 emergency doctor IDs are initialized in the access

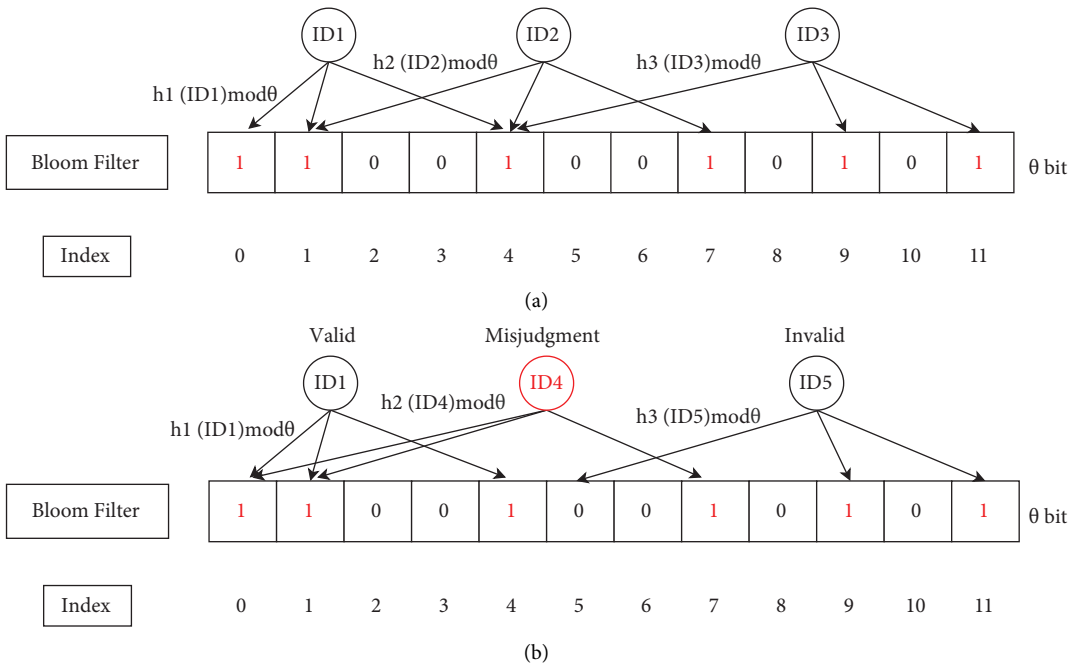


Figure 6: The bloom filter authentication.

control list. If the doctor's input ID is in the access control list, it means that this doctor is authorized. There are too many IDs in the access control list. If the common traversal algorithm is used to query whether the input ID is in the access control list, it is very time-consuming, especially, when multiple doctors use the system to search at the same time. We choose to use the bloom filter to store the doctor's ID, which can effectively shorten the authentication time and speed up the system response. Actually, the bloom filter is a binary array. To judge whether the data is stored in this array is to see if the corresponding bit is 1 or 0 after hash processing; if the corresponding bit is 1, it exists, and if the corresponding bit is 0, it does not exist. The bloom filter authentication process is shown in Figure 6.

(i) Figure 6(a) shows the process of a new ID to be inserted into the bloom filter, which is also the initialization of the access control list. The bloom filter has θ bits. Taking ID_1, ID_2, and ID_3 as an example, assume that $k = 3$ hash functions need to be calculated. Each ID is hashed and modelled to get the corresponding value, and the value of the corresponding index in the bloom filter is then set as 1. Taking $k = 3$ as an example, $h_1(ID_1) \bmod \theta = 0$, $h_2(ID_1) \bmod \theta = 1$, $h_3(ID_1) \bmod \theta = 4$, then the indexes corresponding to ID_1 are set as 0, 1, and 4. In the same way, the indexes corresponding to ID_2 and ID_3 can also be calculated and initialized to 1, respectively.

(ii) Figure 6(b) presents the data query verification process, which determines whether a doctor's ID is legal to access the requesting data. In order to determine whether an ID exists in the access control list, the bloom filter calculates the k hash value of the requesting doctor's ID based on the different hash functions. In this example, the filter calculates the three hash values and checks whether the corresponding bits are all 1. When all the verified bits are 1, then the ID is authorized to access the requesting data. Suppose that there are two doctors ID_1 and ID_5 who request the same patient's record; we verify their query requests one by one. We first calculate three hash values of ID_1 and take their modulus and then find that the three corresponding index values in the bloom filter are 1, showing that ID_1 is a legal doctor to access the data. Similarly, we calculate the three hash indexes for ID_5 and the corresponding hash values are 0, 1, and 1, resulting in the query denial from ID_5.

Following the designed authentication method based on the bloom filter, the time complexity to set and verify the filter for an ID is $O(k)$, regardless of the amount of data. Here, the parameter k is the number of hash functions, which is generally not too large, and we set $k = 5$ in the system implemented in this paper. Comparatively, the time complexity of traditional traversal methods is $O(n)$, where n is the total number of legal IDs in the access control list. Therefore, the proposed design of the bloom filter can accelerate the authentication speed, especially when the access list contains a large number of legal IDs.

Meanwhile, the drawback of the proposed bloom filter is that there exists a certain false positive rate. For example, given an illegal doctor identity ID_4, the calculated indexes are 0, 1, and 7 which are set as 1 for ID_1 and ID_2. In this case, ID_4 will be treated as a legal doctor, resulting in a false positive case happening. The false positive rate decreases with the number of hash functions, and when we set $k = 5$, the false positive rate is approximately 0.01, which is acceptable in our application scene.

4. Security and Privacy Analysis

In this section, we analyze how the proposed system ensures data security and privacy issues against several possible attacks.

Case 1. The proposed PHR system can prevent attackers from illegally obtaining private keys to decrypt data.

Threat model: the PHR system escrows the patient's encrypted private key on the blockchain. The smart contract is invoked to decrypt the patient's private key. Since the smart contract is transparent, the attacker can obtain the private key of the corresponding patient by downloading the smart contract on the blockchain and performing the decryption process locally.

Argument: the proposed system designs an anonymous identity mapping mechanism, which is only visible to the patient himself/herself. The patient stores his/her encrypted private key on the blockchain using an anonymous identity. Even if the attacker downloads the decryption smart contract deployed on the blockchain to obtain the private key, the attacker does not know which patient's data can be decrypted by the obtained private key. Unless the attacker performs countless tests, this operation will consume a lot of computing power. Therefore, the anonymous mapping mechanism can well protect the privacy of patients.

Case 2. The proposed PHR system can resist malicious access.

Threat model: attackers use the PHR system access mechanism that does not require authorization in an emergency situation to obtain the patient's personal health records by maliciously accessing the system, resulting in leakage of patient privacy.

Argument: the proposed PHR system is based on the Hyperledger Fabric, which is a permissioned blockchain. Different from the public blockchain, Fabric does not allow all users to access it and only targets a limited number of third-party organizations to access it. Moreover, our system also creates an ACL. If someone wants to access PHR data, they need to authenticate their identity after entering the system. Besides restricting access, it is necessary to record the access logs of people entering the system. If the attacker breaks through the front barrier and enters the system, all his/her access and operation behaviors will be recorded on the blockchain. Patients can view access logs of their PHR data when they are conscious. If patients find some suspicious behaviors, they can assert their rights through the law.

Case 3. The proposed PHR system can resist the attacker to tamper with data.

Threat model: personal health records include the patients' physical indicators, as well as their previous diagnosis records, medication status, and treatment plan. The attacker's goal is to modify or replace part of the data, thereby causing damage to the PHR data.

Argument: the PHR data that uploads to IPFS is encrypted by symmetric encryption and asymmetric encryption algorithms. The hash value returned by IPFS is stored on the blockchain. Since the blockchain is open and transparent, the attacker can obtain the hash value from the blockchain and query the corresponding file in IPFS, but the file is encrypted. The encrypted file can only be decrypted with the patient's private key. If the attacker wants to modify the content of an encrypted file, he/she needs to reupload the modified file to IPFS to obtain a new hash value and then modify the hash value on the blockchain. Modifying the data on the blockchain needs to create another main chain, which is almost impossible. In addition, for patients and physicians who have permission to update the PHR data, a blockchain-based verification smart contract is deployed that only allows additions and does not allow modification of previously existing content. Therefore, the model effectively resists malicious tampering with the patient's PHR data.

Case 4. The proposed PHR system can resist single-node attacks.

Threat model: the attacker unites multiple computers to launch an attack on a target so that the power of the attack increases exponentially. Or the attacker node is disguised as multiple identities, and the data that needed to be backed up to multiple nodes is deceptively backed up to the same malicious node.

Argument: since the PHR system is a distributed system based on the blockchain, the patient's PHR data is actually stored in IPFS. IPFS is a decentralized storage system that breaks files into countless fragments and stores them in each node. As more and more nodes are added, single-node attacks will have no impact on the system. In addition, in the model proposed in this paper, a single attacker node cannot disguise multiple identities. Firstly, Hyperledger Fabric needs to verify the identity of each node. Secondly, the newly added node in IPFS needs to complete the replication proof and the space-time proof to prove that it is an effective and stable node; otherwise, it will be challenged and punished by the system. Therefore, the model effectively resists single-node attacks.

Through the analysis of the above four cases, we can know that the system proposed in this paper has high security and privacy. It can resist multiple security attacks and ensure the safety of patients' PHR information. The PHR system stores personal private data, including a lot of sensitive information. Therefore, the security and privacy issues of the PHR system have always been a point of concern for users, which is also the reason why the PHR system has not been widely used. The development of this system will further increase the utilization rate of the PHR system.

5. Performance Evaluation

This section mainly introduces the experimental test environment and evaluates the performance of our system according to the results of the experiments. In addition, we compare our experimental results with the systems proposed in other papers, which shows the superiority of the system proposed in this paper.

The experimental test environment is as follows. The host is a machine with Intel(R) Core(TM) i7-6700 CPU, 2.60 GHz, and 8 GB RAM, and the operating system is Ubuntu 16.04LTS, 64 bits. Front-end interaction is implemented by IntelliJ IDEA, Java 1.8, Java security library, Spring boot 2.4.4 [26], and Apache Tomcat 9.0.44. The simulation experiment test tool used is JMeter. The PHR data set used in the experiment comes from [27], and some medical videos come from [28]. For the testing of the blockchain service, the Hyperledger Fabric network is created in Docker [29] environment with Java JDK. The Fabric network includes an endorser node, an orderer node, and two peer nodes.

Experiment 1. Response time for each stage of a single user.

In order to better present the results of the experimental tests and the superiority of the system, we divide the experimental process into three stages, namely, patient uploading PHR data, patient updating PHR data (here, we take patient updating as an example, and the doctor updating is the same), doctor identity authentication, and doctor getting PHR data. Firstly, we test the time spent in three different transaction stages in the case of a single user. The PHR file format uploaded by the patient is pdf and the file size is 2 MB. The format of the updated PHR file is also pdf and the updated file size is 2.4 MB. The experimental results are shown in Table 3. It can be seen that, under the abovementioned conditions, the process of requesting data by the doctor including authentication and getting the patient's PHR data probably only takes about 7.5 seconds.

We compare our experimental results with the experimental results of the system proposed in [13]. The system proposed in [13] is also a PHR access control system based on Hyperledger Fabric in consideration of the emergency situation. This system does not achieve complete privacy protection and response speed optimization. In addition, the access control strategy of this system is role-based access, which means that the emergency doctor has the right to directly access the data at any time. Comparison finds that, in our proposed system, the speed of emergency doctor authentication and getting patient data is greatly improved. The time taken by emergency physicians from entering the system to obtain the complete patient's personal health records has been reduced by approximately 45%. Moreover, the average response time of receiving messages from the emergency contacts in the traditional access control PHR systems [22, 23] is "431.28 s" during simultaneous conversations. It can be seen that, compared to the blockchain-based system proposed in the other literatures and the traditional system, the performance of the system proposed in this paper is superior.

Table 3: The response time of different transaction.

Roles	Transaction	Response time (ms)	Total time (ms)
The patient	Patient upload PHR	3750	3750
	Update patient's PHR	4183	4183
The doctor	Doctor authentication	2498	7547
	Get patient's PHR	5049	

Experiment 2. Comparison of the execution time of each stage when the data size is different under a single user.

For purpose of more intuitively showing the influencing factors that affect the change of the response time of the system at each stage, we conducted the following tests on the system. Taking a single user as an example, the experiment is designed to test the change of response time with the size of the data block in the four transaction processes of patient uploading PHR data, patient update date, doctor identity authentication, and doctor getting data. The data sizes of 64 kB, 128 kB, 512 kB, 2 MB, 8 MB, 32 MB, and 128 MB are tested, respectively, to get the response time at different transaction stages. For the update stage of patient data, the size of the updated data is slightly larger than the corresponding original uploaded data. The size of the updated data only affects the time to update the data on the blockchain and does not affect the time to verify whether the data is modified, because the part verified for the updated data is always the same part as the original uploaded data. Moreover, the update part of the experiment is based on the successful update as an example.

The experimental results are shown in Figure 7. Figure 7(a) shows that the time for patients to upload and update PHR data rises with the increase of the PHR data size. When the data size is the same, the time of updating the PHR data stage almost and always exceeds the time of uploading stage. The time gap between the two stages increases with the size of PHR data, because the difference is the time that it takes for the smart contract to verify whether the updated data follows the principle of only being added but not changed, and the time of uploading data to the blockchain and updating the data in the block of the blockchain is almost the same. The verification part uses the file comparison algorithm, so the verification time rises with the increase of the PHR data size. In Figure 7(b), we can observe that the response time for doctor identity authentication remains stable basically and does not change with the size of the PHR data uploaded by the patient. This shows that the response time of the doctor authentication stage is independent of the size of the PHR data uploaded by the patient. Figure 7(c) shows that the response time of the doctor getting the PHR data stage rises with the increase of the data size. Meanwhile, the response time increases greatly with the size of the data changing greatly. When comparing the line graph of the patient uploading the PHR data stage in Figure 7(a) and the line graph in Figure 7(c), the following conclusions can be drawn. When the size of data is small, the doctor getting PHR data stage takes longer than the uploading stage, because the doctor getting PHR data stage includes multiple processes of decrypting the patient's

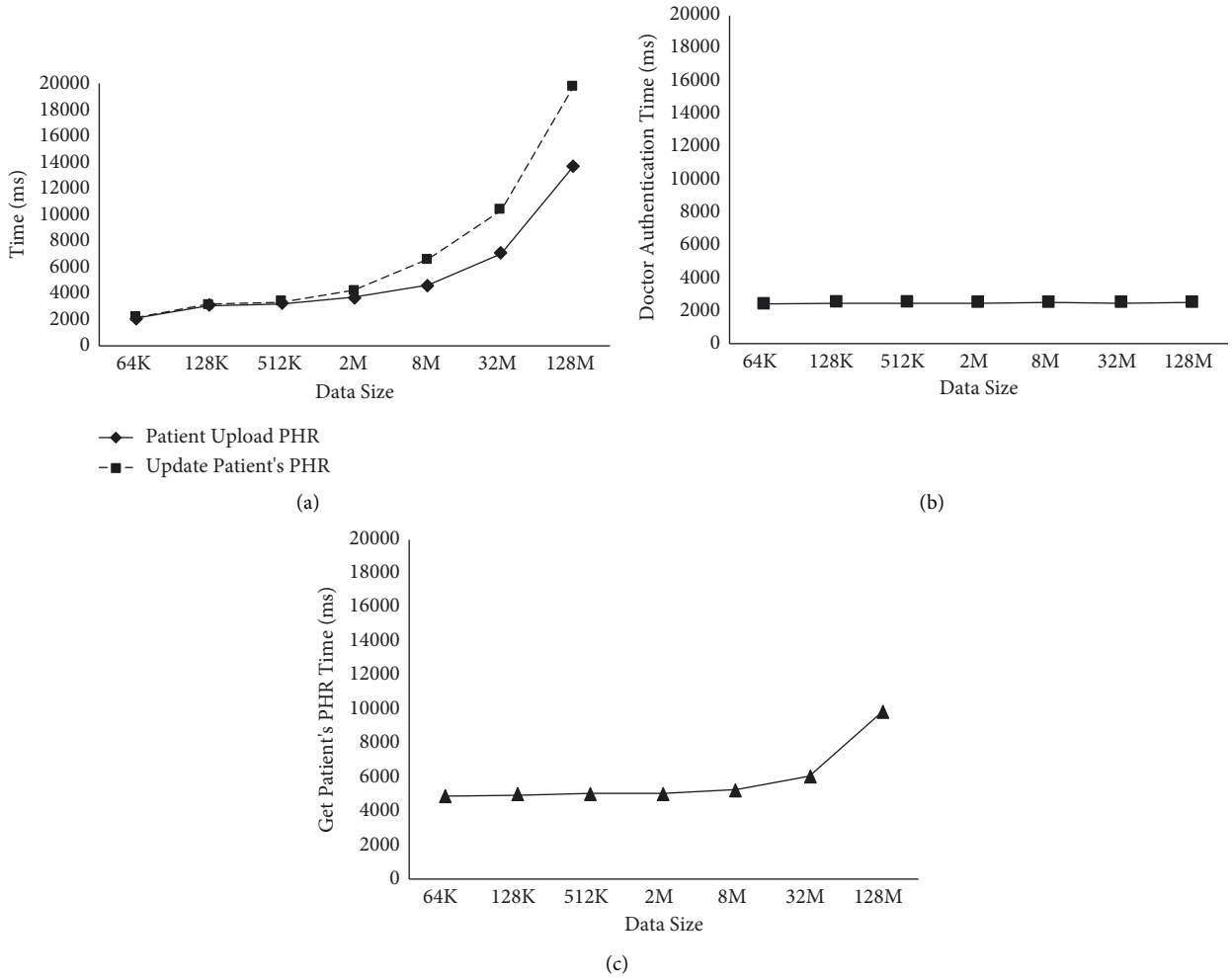


Figure 7: The response time of different data size.

private key, reencrypting the data by the smart contract, and decrypting the data by doctor's private key. When the size of data is large, the patient uploading data stage takes longer than the doctor getting PHR data stage, because the data needs to be encrypted in the uploading stage. The larger the data block, the longer the encryption time. Therefore, it can be known through experiments that the data size is one of the important factors affecting the response speed of the system.

Experiment 3. Comparison of doctor authentication time with and without bloom filter.

To verify the optimization of the response speed of the system designed in this paper, we adopted the following experimental scheme. We compare the time to authenticate the doctor's identity using the bloom filter and using the traditional traversal method without using the bloom filter in the case of different numbers of users. We simulate and test the response time of doctor identity authentication when the number of concurrent users is 20, 40, 60, 80, 100, 120, 140, and 160. Since the emergency system does not have very large-scale concurrency, the maximum number of concurrent users that we choose is only 160.

The experimental results are shown in Figure 8. We can see that as the number of users increases, the advantages of using the bloom filter become more and more obvious. According to the experimental results, when the number of users is 20, the authentication time of the system using the bloom filter is shorter than that of the unused system. As the number of concurrent users gradually increases, the difference between the identity authentication time of doctors using the bloom filter system and the unused system is getting bigger and bigger. Comparing the experimental results horizontally, we can find that the advantages of using the bloom filter gradually become apparent as the number of concurrent users increases.

It can be seen from three experiments that the performance of the system proposed in this paper is better than some systems proposed in the other literatures, so the optimization method that we proposed is effective.

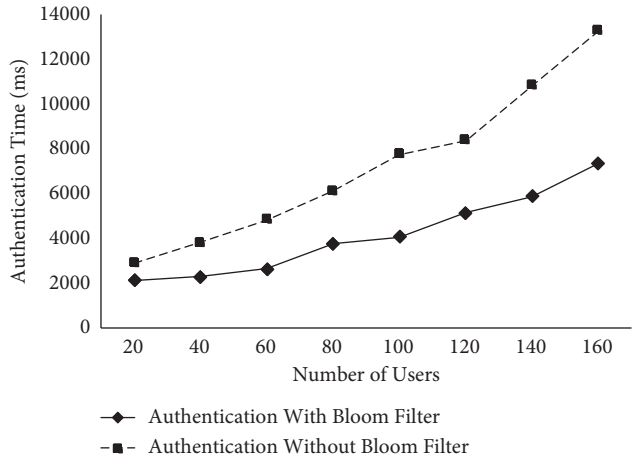


Figure 8: The authentication time comparison.

6. Conclusion

In this work, a personal health record system based on blockchain is proposed to protect the privacy of patients and improve the access control efficiency of their records in emergency situations. The proposed solution uses blockchain and IPFS to store health record data, where IPFS stores encrypted complete data and blockchain stores the data hash to mitigate the data storage cost issue. We use cryptography technologies such as symmetric encryption, asymmetric encryption, and reencryption to ensure the privacy of patient PHR data. We propose private key escrow and access control list to enable emergency doctors to access in emergency situations. In addition, we optimize the response speed of the system by designing a QR code and bloom filter so that emergency doctors can efficiently access patients' records even when they are conscious. The proposed system is evaluated by comparing it with the existing ones in the literature, and the experimental results show that our system can decrease the authentication time by 45%, demonstrating the efficiency of the proposed system.

In future work, we plan to apply the proposed system in a realistic application scenario and collect the runtime performance data to do further evaluations.

Data Availability

Data used in this study is available at https://catalog.data.gov/dataset/va-personal-health-record-sample-data.

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 National Key Research and Development Plan (Grant no. 2018YFB1800701), National Natural Science Foundation of China, under Grant nos. 62172085, 61972105, U20B2046, and 62032013, Key-Area Research and Development Program of Guangdong Province (no. 2020B0101090005), and China-Singapore International Joint Research Institute, Guangzhou, China.

References

[1] P. C. Tang, J. S. Ash, D. W. Bates, J. M. Overhage, and D. Z. Sands, "Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption," *Journal of the American Medical Informatics Association*, vol. 13, no. 2, pp. 121–126, 2006.

[2] D. B. Lafky and T. A. Horan, "Prospective personal health record use among different user groups: results of a multi-wave study," in *Proceedings of the 41st Hawaii International International Conference on Systems Science Waikoloa*, pp. 1–9, IEEE Computer Society, Hawaii, HI, USA, January 2008.

[3] J. Israelson and E. C. Cankaya, "A hybrid web based personal health record system shielded with comprehensive security," in *Proceedings of the 45th Hawaii International International Conference on Systems Science Maui*, pp. 2958–2968, IEEE Computer Society, Maui HI, USA, January 2012.

[4] C. Wang, X. Xu, D. Shi, and W. Lin, "An efficient cloud-based personal health records system using attribute-based encryption and anonymous multi-receiver identity-based encryption," in *Proceedings of the 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing Guangdong*, pp. 74–81, IEEE Computer Society, China, 2014.

[5] Y. Liu, W. Yu, Z. Ai, G. Xu, L. Zhao, and Z. Tian, "A blockchain-empowered federated learning in healthcare-based cyber physical systems," *IEEE Transactions on Network Science and Engineering*, p. 1, 2022.

[6] J. Benet, "IPFS - content addressed, versioned, P2P file system," *CoRR*, vol. 3561, pp. 1–11, 2014.

[7] J. Chen, C. Zhang, Y. Yan, and Y. Liu, "FileWallet: a file management system based on IPFS and hyperledger fabric," *Computer Modeling in Engineering and Sciences*, vol. 130, no. 2, pp. 949–966, 2022.

[8] L. Zhang, Y. Ye, and Y. Mu, "Multiauthority access control with anonymous authentication for personal health record," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 156–167, 2021.

[9] S. Murphy, "The advanced encryption standard (AES)," *Information Security Technical Report*, vol. 4, no. 4, pp. 12–17, 1999.

[10] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[11] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, and A. D. Caro, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference Porto Portugal*, pp. 1–30, ACM, Porto, Portugal, April 2018.

[12] M. N. Huda, S. Yamada, and N. Sonehara, "Privacy-aware access to patient-controlled personal health records in emergency situations," in *Proceedings of the 3rd International Conference on Pervasive Computing Technologies for Healthcare*, pp. 1–6, IEEE, London, UK, April 2009.

[13] A. R. Rajput, Q. Li, M. Taleby Ahvanooey, and I. Masood, "EACMS: emergency access control management system for personal health record based on blockchain," *IEEE Access*, vol. 7, pp. 84304–84317, 2019.

[14] B. Liu and J. Xu, "Access control based on proxy Re-encryption technology for personal health record

systems,"vol. 12239, pp. 411–421, in *Proceedings of the Artificial Intelligence and Security - 6th International Conference, ICAIS*, vol. 12239, pp. 411–421, Springer, Hohhot, China, July 2020.

[15] H. X. Son, H. T. Le, N. Q. T. Tang, H. N. D. Huy, N. Duong-Trung, and H. H. Luong, "Toward a blockchain-based technology in dealing with emergencies in patient-centered healthcare systems,"vol. 12605, pp. 44–56, in *Proceedings of the Mobile, Secure, and Programmable Networking - 6th International Conference, MSPN*, vol. 12605, pp. 44–56, Springer, Paris, France, October 2020.

[16] H. M. Hussien, S. M. Yasin, N. I. Udzir, and M. I. H. Ninggal, "Blockchain-based access control scheme for secure shared personal health records over decentralised storage," *Sensors*, vol. 21, no. 7, p. 2462, 2021.

[17] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

[18] T. Bhatia, A. K. Verma, and G. Sharma, "Secure sharing of mobile personal healthcare records using certificateless proxy re-encryption in cloud," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 6, pp. e3309–e3321, 2018.

[19] S. Wang, D. Zhang, and Y. Zhang, "Blockchain-based personal health records sharing scheme with data integrity verifiable," *IEEE Access*, vol. 7, no. 99, pp. 102887–102901, 2019.

[20] T. T. Thwin and S. Vasupongayya, "Blockchain-based access control model to preserve privacy for personal health record systems," *Security and Communication Networks*, vol. 2019, no. 5, pp. 1–15, 2019.

[21] F. Aljumah, R. H. M. Leung, M. Pourzandi, and M. Debbabi, "Emergency mobile access to personal health records stored on an untrusted cloud,"vol. 7798, pp. 30–41, in *Proceedings of the Health Information Science - Second International Conference, HIS*, vol. 7798, pp. 30–41, Springer, London, UK, March 2013.

[22] P. Thummavet and S. Vasupongayya, "A novel personal health record system for handling emergency situations," in *Proceedings of the International Computer Science and Engineering Conference, ICSEC Silpakorn Univ*, pp. 266–271, IEEE, Nakhon Pathom, THAILAND, September 2013.

[23] P. Thummavet and S. Vasupongayya, "Privacy-preserving emergency access control for personal health records," *Maejo International Journal of Science and Technology*, vol. 9, no. 1, pp. 108–120, 2015.

[24] M. M. Madine, K. Salah, R. Jayaraman et al., "Fully decentralized multi-party consent management for secure sharing of patient health records," *IEEE Access*, vol. 8, no. 99, pp. 225777–225791, 2020.

[25] V. Koufi, F. Malamateniou, and G. Vassilacopoulos, "A personal health record system for emergency case management,"vol. 127, pp. 83–96, in *Proceedings of the Biomedical Engineering Systems and Technologies - Third International Joint Conference, BIOSTEC*, vol. 127, pp. 83–96, Springer, Valencia, Spain, January 2010.

[26] A. C. Ríos, A. Boyko, A. Nesterov, A. Clement, and C. Walls, *Spring boot*Spring Inc, Tiruppur, TN, India, 2022.

[27] D. V. O. Usdatagov, Ed., *VA Personal Health Record Sample Data - Data.Gov*, Department of Veterans Affairs, Coimbatore, TN, India, 2022.

[28] M. Roger Seheult and P. C. Kyle Allred, Eds., *CME MedCram - Best Medical Lectures and Medical Videos*, MedCram, LLC, California, CA, USA, 2022.

[29] S. Johnston, J. L. de Morlhon, M. Carter et al., Eds., *Docker*, Docker Inc, Palo Alto, CA, USA, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 9062615, 12 pages
https://doi.org/10.1155/2022/9062615

Research Article

A Multiuser Ciphertext Search Scheme Based on Blockchain and SGX

Lianhai Wang,[1] **Fengkai Liu**,[1] **Lingyun Meng**,[1] **Wei Shao**,[1] **Shujiang Xu**,[1] **Shuhui Zhang**,[1] **and Donghui Huang**[2]

[1]*Qilu University of Technology (Shandong Academy of Sciences), Shandong Computer Science Center (National Supercomputer Center in Jinan), Shandong Provincial Key Laboratory of Computer Networks, Jinan 250014, China*
[2]*Rizhao Port, Group of Shandong Port, Rizhao 276800, China*

Correspondence should be addressed to Lianhai Wang; wanglh@sdas.org

Received 31 July 2022; Accepted 6 September 2022; Published 23 September 2022

Academic Editor: Yinbin Miao

To ensure the security of data, more and more users encrypt data for storage, which makes the high-efficiency ciphertext search problem in the context of cloud storage a research hotspot. Existing solutions still suffer from many vexatious problems, such as the need to maintain complex index structures and the unsatisfactory application of homomorphic schemes. To solve the above problems, this paper proposes a multiuser ciphertext search scheme based on blockchain and SGX. Our scheme uses blockchain and SGX to protect keywords and data privacy and complete decryption and keywords search of ciphertext data which does not need pregenerated indexes or preselected keywords. Second, for a multiuser scenario, a smart contract is designed to verify authorization requests and manage multiple authorized users. Finally, we give security analysis, function comparison, and performance analysis to prove the security and feasibility of our scheme. Experiments show that our scheme has effectively met practical requirements.

1. Introduction

With the rapid development of Internet of Things (IoT) technology, the amount of data generated by various applications has increased dramatically. To solve the problem of massive data storage, many enterprises and individuals choose to outsource data storage to cloud servers. Cloud storage can not only reduce local storage costs of users but also allow them to download and use the outsourced data regardless of device, access times, and geographical restrictions. However, it also brings problems such as data leakage and security risks. In 2021, data leakage incidents occurred frequently, which brought huge losses to enterprises and awakened society to the importance of data security. Therefore, privacy protection, data integrity, and sustainable services in the context of cloud storage have become major issues that cannot be ignored.

To prevent illegal data access by servers or unauthorized users, data should not be stored in plaintext. Users should encrypt data before uploading individual data to a cloud server. However, the commonly used data encryption scheme will limit the ability of the cloud server to process user access requests. In other words, the original search function of the cloud server will be invalid due to encryption, making the data retrieval a very difficult task. If a user wants to query data containing a certain keyword, he needs to download the encrypted data and then decrypt it for content search. The method is utterly inefficient since it requires extra space overhead and brings a poor experience to the user. Hence, the ciphertext search problem in the context of cloud storage has become a hot research topic in academia. In 2000, Song et al. [1] proposed a method that can perform a search on ciphertexts, namely, the symmetric searchable encryption (SSE) technology. At present, the research of SSE

technology for data retrieval has been developed to a certain extent. A typical SSE model is shown in Figure 1. A user first generates the index of a file from its keywords. A specific SSE algorithm is used to encrypt the file before it is uploaded to the cloud server. To fulfill a search, a data requester sends a Trapdoor to the cloud server. The cloud server searches the ciphertext for the Trapdoor and returns the search results. However, such models cannot guarantee data security when cloud servers are dishonest or compromised (centralization problems). To solve this problem, some schemes improved the security of data cloud storage by combining blockchain technology with SSE technology.

Blockchain [2] is a distributed database that creates a fully trusted environment between unfamiliar individuals without the need for third-party trust endorsements. Moreover, blockchain combined with cryptography technology can ensure transaction traceability, irreparable modification, and nonrepudiation. It is widely used for secure data sharing and large-scale collaborative computing and has been regarded as a powerful tool to solve data security and privacy issues in the context of cloud storage. At present, some schemes [3–10] combine blockchain and SSE technology. A user first stores encrypted data in Interplanetary File System (IPFS) and then sends the generated indexes and Trapdoor to smart contracts. Smart contracts can perform keyword search operations instead of cloud servers and finally return the search results to users, avoiding the centralization problem of cloud servers. However, the existing solutions mainly focus on keyword-based search. Users need to share keyword sets and encryption keys, which leads to many restrictions on the selection of keywords for users. The risk of data leakage by direct data sharing can also not be neglected. Since the search effect depends on the correlation between keywords and files, it is difficult to meet the data retrieval requirements for ciphertext in cloud storage systems. Furthermore, in the data search phase, a large amount of on-chain overhead can be costly for smart contracts to perform search operations. We need to consider moving data search operations to the off-chain to process.

Homomorphic encryption (HE) technology is an encryption method that allows direct operation on ciphertexts [11]. Some researchers use HE technology for ciphertext retrieval research [12–17]. Users first perform homomorphic encryption on files, under the premise of effectively protecting the privacy of users' sensitive data, the cloud server is entrusted to directly perform addition and multiplication isomorphic operations on the ciphertext data, and the result is equivalent to the operation on the plaintext. However, such schemes suffer from a large computational overhead problem. Therefore, the homomorphic encryption scheme is difficult to apply in practice.

To solve the above problems, we propose a multiuser ciphertext search scheme based on blockchain and SGX. Specifically, to ensure the privacy of user data and solve the inability of smart contracts to be applied to complex computing scenarios, we combine blockchain and SGX to design a new ciphertext search model. The combination of blockchain and SGX will encounter challenges in data interaction and result traceability. But we solve the challenge of data interaction through on-chain contract storage and off-chain call acquisition. And then we record the search results on the blockchain to solve the traceability challenge of the search calculation result. In addition, for a multiuser scenario, we also design a smart contract to authorize and manage users. In summary, the main contributions of this paper are shown as follows:

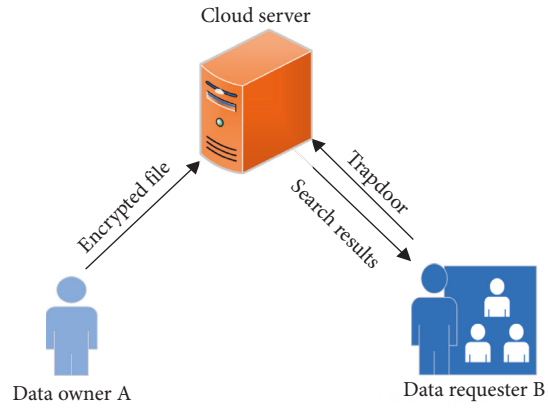


Figure 1: A typical searchable encryption model.

(1) First, we propose a multiuser ciphertext search scheme based on blockchain and SGX. In this scheme, we use blockchain and SGX to protect keywords and data privacy and complete decryption and keywords search of ciphertext data which does not need pregenerated indexes or preselected keywords.

(2) Second, for a multiuser scenario, a smart contract is designed to verify user authorization requests and manage multiple authorized users.

(3) Finally, we give security analysis, function comparison with other schemes, and performance analysis to prove the security and feasibility of our scheme. Experiments show that our scheme has effectively met practical requirements.

1.1. Organization. The rest of this paper is organized as follows: Section 2 introduces related works. Section 3 presents preliminaries. Section 4 describes the system model, threat model, and design goals of the scheme. Section 5 presents the specific construction and protocol of the scheme. Section 6 delivers security analysis and functional comparisons. In Section 7, we test and analyze the functionality of the scheme. Finally, Section 8 concludes the paper.

2. Related Works

2.1. Searchable Encryption Schemes. At present, as a research hotspot in the field of cloud storage security, searchable encryption has made great progress. In 2000, Song et al. [1] proposed a symmetric searchable encryption scheme (SSE). SSE scheme [18, 19] has high encryption efficiency, but its

key management is more complicated in the data sharing phase. To solve this problem, Boneh et al. [20] proposed a public key searchable encryption scheme supporting keyword search, and to narrow down the keyword search, searchable encryption schemes [21–25] supporting multiple keywords were proposed. Xia et al. [26] proposed a multikeyword sorting search scheme supporting dynamic updates. Li et al. [27] proposed a searchable encryption scheme using a fixed server to verify the user's identity in an e-mail sending and receiving environment, which improves the security requirements of the scheme. Yang et al. [28] proposed a search scheme that supports both multikey search and semantic sorting. Wang et al. [29] used each leaf node in a Merkle tree to store the MAC corresponding to the index and kept the root node as evidence locally. However, this scheme is implemented in a single-user model, which includes only two entities, the data owner and the server. Chen et al. [30] implemented forward and backward security in their scheme, but this scheme does not address the authorization problem well in a one-to-many model. The above schemes provide users with search results that satisfy the actual needs, but they fail to satisfy the needs of multiple users for data search and fail to achieve data access control. In addition, cloud encrypted data faces centralization problems such as server-side untrustworthiness and tampering of stored data.

In order to achieve multiuser access to data, in combination with attribute-based encryption (ABE), searchable encryption schemes can achieve keyword search while enabling fine-grained access control of encrypted files. Yin et al. [31] proposed a ciphertext policy attribute-based (CP-ABE) searchable encryption scheme, which has a high possibility of causing the server to return search results containing a large amount of irrelevant content and waste network bandwidth. Lin et al. [32] proposed an attribute set-based Boolean keyword search scheme, which can realize fine-grained access control and Boolean keyword search over encrypted personal health records (PHR). Zhang et al. [33] proposed a practical CP-ABE scheme, which offers users revocation and attribute update. The ciphertext size and decryption cost grow with the complexities of access policies. Mao et al. [34] gave the generic construction of Chosen-Plaintext Attack (CPA) secure CP-ABE scheme with verifiable outsourced decryption. Sun et al. [35] proposed a verifiable attribute-based ciphertext retrieval scheme, which allows multiple owners to encrypt and outsource their data to the cloud server independently. The scheme supports user attribute write-off and can verify the results returned by the server in a many-to-many scenario, but the scheme has large storage overheads. Miao et al. [36] proposed a secure multiauthority CP-ABKS (MABKS) system to avoid having performance bottleneck at a single point in cloud systems.

2.2. Searchable Encryption Schemes Based on Blockchain. With the continuous maturity of blockchain technology, some schemes have introduced blockchain technology into searchable encryption to solve the centralization problems faced by encrypted data in the cloud such as server-side untrustworthiness and tampering of stored data. Zheng et al. [37] proposed a blockchain-enabled public key encryption scheme with multikeyword search (BPKEMS), which supports file updates. Moreover, a smart contract is used to ensure the fairness of transactions between the data owner and user without introducing a third party. Chen et al. [30] proposed a public key searchable encryption scheme in Vehicle Social Network (VSN), which replaces the original cloud server with a smart contract in the blockchain. Li and Wang et al. [38, 39] studied searchable encryption in a cloud environment, and two schemes reduced the search time under a large number of keywords. Jiang et al. [3] proposed a search scheme that supported multiple keywords and reduced the computation of the scheme and improved the efficiency of the scheme. Yang et al. [4] proposed a searchable encryption scheme in a shared electronic medical record scenario. The scheme stores the ciphertext of electronic medical records in the cloud server and the keyword ciphertext in the blockchain. Zhang et al. [5] introduced a dynamic accumulator algorithm into a blockchain searchable encryption scheme to improve the cryptographic search performance of the scheme. Guo et al. [6] designed a dynamic searchable encryption scheme based on the blockchain and used smart contracts in the blockchain to implement the verifiable function. Poongodi et al. [7] used the blockchain to design a trusted architecture using encryption and hashing methods to achieve reliable keyword search. Searchable encryption schemes generally suffer from high computational and storage overhead. Xu et al. [8] proposed a postquantum public key searchable encryption scheme on blockchain (PPSEB) for E-healthcare scenarios, which utilized a lattice-based cryptographic primitive to ensure the security of the search process and introduced blockchain technology to solve the problem of third-party untrustworthiness in the search process. Fu et al. [9] and Liu et al. [10] proposed a blockchain-based searchable encryption scheme in which the blockchain is used to store secure indexes and deploy smart contracts to perform the search of ciphertext files. All the above schemes are index-based ciphertext search schemes, which can effectively protect users' data security because the search object is encrypted data. However, the semantic relationship of words is lost, and keyword search operation cannot be performed on ciphertext, so the index corresponding to encrypted data needs to be generated in advance, and a complex index structure needs to be maintained. In practical applications, users cannot search data beyond the predefined indexes and are restricted in the selection of keywords. In addition, a large amount of on-chain overhead is required in the data search phase, and a large amount of data needs to be considered to be processed off-chain.

2.3. Ciphertext Search Schemes Based on Homomorphic Encryption. In recent years, in order to improve the accuracy and security of ciphertext retrieval, HE technology has attracted the attention of many researchers. The user first performs homomorphic encryption on plaintext data. On the premise of ensuring data privacy, the cloud server is

entrusted to perform homomorphic operation on ciphertext data directly, and the search result is equivalent to the operation on plaintext. In 1978, the HE scheme was first proposed by Rivest et al. [11]. In 2009, Gentry et al. [12] implemented homomorphic encryption theoretically. And then someone proposed the DGHV scheme [13] and the GSW13 scheme [14] in 2010 and 2013, where the former implemented an integer-based fully homomorphic encryption based on the approximate maximum convention number problem and the latter proposed the first identity-based homomorphic encryption scheme based on the error learning problem, to effectively guarantee the security of outsourced data and satisfy users to efficiently retrieve data stored in the cloud. In 2018, Fu et al. [15] proposed CRSHE: a new ciphertext retrieval scheme based on homomorphic encryption, which effectively solves the problems of privacy leakage of retrieved keywords and nonsupport of homomorphic encryption and improves the search efficiency and accuracy. In 2020, Han et al. [16] proposed a homomorphic encryption-based full-text retrieval scheme for cloud storage, which combines integer vector encryption technology with vector space model and is applied in full-text retrieval in third-party untrusted cloud storage. In 2021, Liu et al. [17] proposed a homomorphic encryption-based keyword search scheme in cloud servers, which has higher accuracy compared with the traditional ciphertext search scheme. Such schemes can directly perform computer on ciphertext data, but they require larger computational overhead or lower search efficiency, which is difficult to apply in practice.

3. Preliminaries

3.1. Blockchain Technology. The blockchain is a chained data structure formed by connecting multiple data blocks through a hash function, as shown in Figure 2. It realizes data verification, sharing, computing, storage, and other functions through a consensus mechanism. The blockchain provides a distributed trust ledger for each participant, and each node or user maintains and stores the same ledger to ensure that all users and nodes in the corresponding blockchain are completely consistent. A smart contract on a blockchain is an automatically running program that automatically performs some functions driven by time or events. It is a decentralized program code deployed in the blockchain to execute. Therefore, the smart contract provides programmability for the blockchain. The main high-level languages for writing smart contracts on Ethereum are Solidity, Serpent, and LLL; it is implemented by storing it compiled into bytecode.

3.2. SGX. The wide application of blockchain technology enables applications to ensure the security of data on the chain. However, in the off-chain processing of data, blockchain technology cannot guarantee its security, so off-chain data processing requires a trusted execution environment based on hardware or software.

The solution adopted in this paper is to rely on the software protection extension Software Guard Extensions (SGX) [40] launched by Intel. The SGX is an extension of the Intel instruction set architecture, which provides hardware-level security for the operation of the program, rather than based on external software. It allows the application to open up a protected and trusted executable area in the memory, called an Enclave. The Enclave provides integrity protection for the programs. If someone attempts to access the Enclave outside the safe area, he will be rejected. After the data in the Enclave is transmitted to the nonsafe area through special encryption, even if other machines get the encrypted data, encrypted data cannot be decrypted, which ensures the correctness and confidentiality of the data.

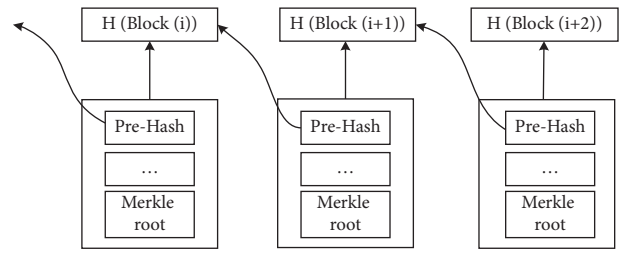


Figure 2: Blockchain structure.

4. Problem Formulation

In this section, we describe in detail the system model, threat model, and design goals in the scheme and design a multiuser ciphertext search scheme based on blockchain and SGX.

4.1. System Model. In this scheme, the proposed system model mainly contains five entities: data owner A, data requester B, data storage DS, blockchain BC, and query node S, as illustrated in Figure 3.

4.1.1. Data Owner A. The data owner is an entity with a large amount of data but limited resources. He uses a key to encrypt personal data and upload it to the data storage. And data authorization requests and authorized users are verified and managed; different data owners allow different query nodes to perform data searches on the ciphertext.

4.1.2. Data Requester B. The data requester is the entity that requests the ciphertext data to search. When the data requester wants to search for keywords, he needs to send a data authorization request to data owner A and finally obtains the plaintext data after decryption according to the encrypted search result.

4.1.3. Data Storage DS. Data storage is a kind of platform that provides distributed storage service for data owner A. It has huge storage space and powerful computing power; however, it is not trustworthy. When uploading ciphertext data, it first verifies the integrity of the data, stores the related data after successful verification, and returns the corresponding storage hash address.

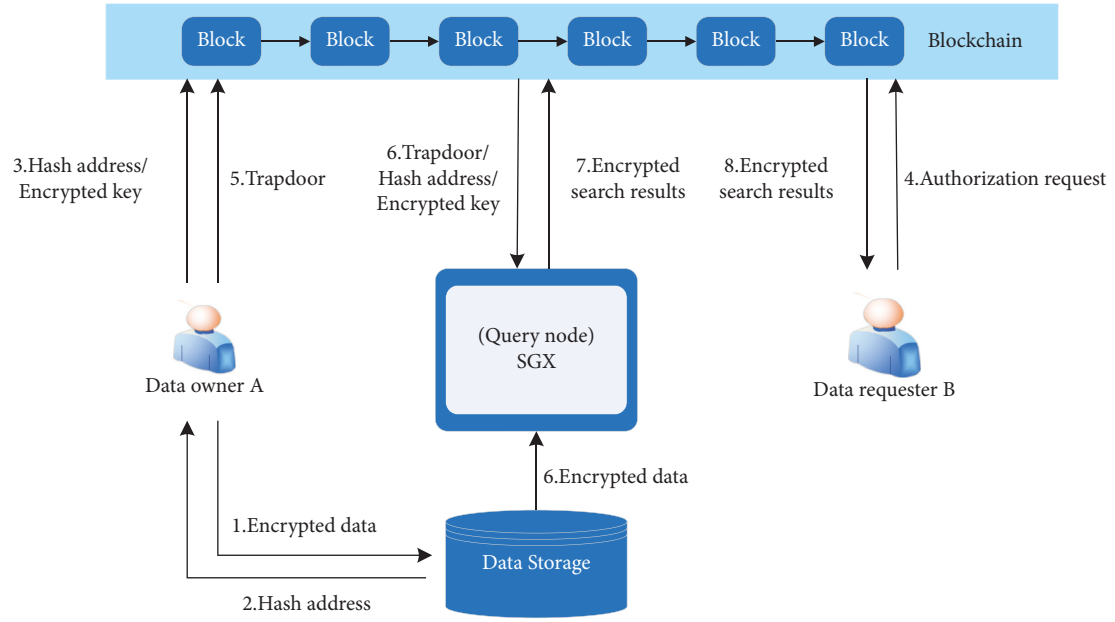


Figure 3: System model.

4.1.4. Blockchain BC. It is a public chain composed of data owners, data requesters, and query nodes. Anyone can join this public blockchain to view or publish transactions. It is mainly used for data storage, transaction recording, and smart contract deployment.

4.1.5. Query Node S. The query node is a registered node on the blockchain. It has its own corresponding SGX trusted execution environment and can take advantage of its own SGX trusted hardware. SGX is a trusted and independent execution environment that exists independently of an untrusted operating system, providing a safe and confidential space for private data and sensitive computing in an untrusted environment.

4.2. Threat Model. In this scheme, data request B is considered semicredited. Only after the verification of the authorization request, data request B can obtain the search permissions for encrypted data. We assume that the key storage of data owners A and data requester B is safe, not attacked by attackers, and all search tasks are performed in SGX Enclave. Secondly, external attackers steal the transmitted data transmitted through public channels and hope to read or modify the data of data owner A. In addition, if the storage data is not accessed for a long time, it may cause loss and other conditions.

4.3. Design Goals. In this scheme, our main design goals are as follows:

(1) Data privacy: In this scheme, since the personal data of data owner A is very sensitive, data security protection is necessary, so in this scheme, the data is encrypted and uploaded to the data storage DS to store.

(2) Storage integrity: In this scheme, the data storage DS stores the encrypted data only when the integrity of the data is verified.

(3) User access control: In this solution, when data requester B wants to search the personal data of data owner A, he needs to send an authorization request to data requester A, and the search permission can only be obtained after the authorization is successful.

(4) Privacy of keywords: In this scheme, due to the encrypted transmission of query keywords, other entities cannot obtain any information about keywords through the ciphertext of the keyword.

5. Our Scheme

In this subsection, we introduce the system construction and protocol of this scheme in detail. This scheme includes six phases: system initialization phase, data processing phase, data storage phase, user authorization phase, data search phase, and data decryption phase. And Table 1 gives some important notations and descriptions used in the following paper. Details are as follows.

5.1. Scheme Construction

5.1.1. System Initialization Phase. Based on the security parameters, the keys and system parameters are generated in the following ways, as follows.

We choose the secure SHA-256 hash algorithm hash (m) and RSA signature algorithm $\text{Sign}_{\text{SK}}(m)$, where $\text{Sign}_{\text{SK}}(m)$ denotes that SK signs m. The system calls the ECC algorithm to generate the key pair $(\text{PK}_o, \text{SK}_o)$ and $(\text{PK}_u, \text{SK}_u)$ for data owner A and data requester B, respectively, and then calculates their Ethereum network addresses addr$a$ and addr$b$. The data owner A uses the DES algorithm to generate a

Table 1: Notations and descriptions.

Notations	Descriptions
(PK_o, SK_o)	Public/secret key pair of A
(PK_u, SK_u)	Public/secret key pair of B
(PK_{sgx}, SK_{sgx})	Public/secret key pair of Enclave
K	Symmetric encryption key of A
M	A plaintext data collection
C	A ciphertext data collection
H	A ciphertext data hash collection
Set	The outsourced collection
CT	The data authorization request
Tr	The Trapdoor

symmetric key k, generating the Enclave key pair (PK_{sgx}, SK_{sgx}) of the SGX corresponding to the query node. System parameters is para $= \{\text{hash}(m), \text{Sign}_{SK}(m), \text{addr}\,a, \text{addr}\,b\}$.

5.1.2. Data Preprocessing Phase. As a registered user of the blockchain, data owner A first divides M into equal-sized data m_i $(i = 1,2, \ldots, n)$ and uses a symmetric key k to encrypt a plaintext data collection $M = \{m_1, m_2, \ldots, m_n\}$ and generate a ciphertext data collection $C = \{c_1, c_2, \ldots, c_n\}$, where the ciphertext data $c_i = Enc_k(m_i)$, and then generate a ciphertext data hash collection $H = \{h_1, h_2, \ldots, h_n\}$, where the ciphertext data hash value $h_i = \text{hash}(c_i)$, and finally write search program P.

5.1.3. Data Storage Phase. Data owner A uses his own private key SK_o to sign PK_o and H and P to generate $r = \text{sign}_{SK_o}(PK_o \| H \| P)$ and send the outsourced collection Set $= (PK_o \| C \| H \| P \| r)$ to the data storage DS. After the data storage DS receives the Set, first, it verifies the validity of the signature r and the integrity of the ciphertext data collection C. If the verification passes, store $H \| C \| P$ in the data storage DS, and then data storage DS returns the corresponding storage hash address addr $= \{\text{addr}_1, \text{addr}_2, \ldots, \text{addr}_n\}$. Otherwise, the verification fails, and the outsourced collection set needs to be reuploaded.

The data owner A uses the SGX remote authentication mechanism to authenticate the identity information of the SGX Enclave of the query node. After the authentication is passed, the query node obtains the search program P from the data storage DS and then installs and deploys it in the Enclave. The data owner A uses the PK_{sgx} encryption key k, organizes the hash value collection H, and stores the hash address addr and other information, generates a timestamp $ts1$, and finally records the txdata in the blockchain. Algorithm 1 describes the process of data owner A's data transaction txdata generation.

5.1.4. User Authorization Phase. If data requester B wants to search the ciphertext data of data owner A, he first uses private key SK_u to encrypt his public key PK_u and query keyword wCT $= \text{Enc}_{SK_u}(PK_u \| w)$ and then sends the data authorization request CT in the form of a transaction to data owner A. After the data owner A receives the data authorization request CT, he first uses addr$b$ to query whether the legal user list in the smart contract contains this user. If the query is successful, it means that the data requester B can obtain the search permission; if the query fails, the data owner A then decrypts the CT using the data requester B's public key PK_u and then performs the Keccak-256 hash operation on the public key PK_u of the data requester B, truncates the last 20 bytes into the string $R1$, and finally calls the smart contract to $R1$ matches addr$b$. If it returns 1, it means that the authorization request is verified, and the data owner A adds the data requester B to the list of legal users through the smart contract; if it returns 0, it means that the data requester B cannot obtain the search permission. In addition, data requester B can also be removed or revoked from the list of legal users through the smart contract.

After the verification is passed, the data owner A uses k to encrypt w to generate a Trapdoor $\text{Tr} = \text{Enc}_k(w)$ and sends Tr to the query node through the blockchain.

5.1.5. Data Search Phase. The Intel SGX extension employs two data sealing schemes: the safe zone strategy (MERNCLAVE) policy and the sealed strategy (MRSIGNER) policy. In this scheme, MRSIGNER is used to query nodes and generate public and private key pairs (PK_{sgx}, SK_{sgx}) in the Enclave, the private key obtains key (SK_{sgx}) through the MRSIGNER policy and outputs it to the nonsecure area, and the public key is explicitly output to the nonsecure area and uploaded to the chain. The query node obtains the Tr sent by the data owner A through the blockchain and performs data decryption and keyword search operation in SGX trusted execution environment. The query node sends the final search results to the blockchain.

Specifically, the SGX trusted execution environment corresponding to the query node first performs integrity verification on the ciphertext data and executes step (1). The SGX trusted execution environment corresponding to the query node performs decryption and keyword search operations and executes steps (2)–(6). The query node sends the encrypted search result to the data requester B and executes step (7), taking the search for ciphertext data c_i as an example. Algorithm 2 describes the process of decryption and keyword search;

(1) The query node first uses the MRSIGNER policy of the corresponding SGX, decrypts key(SK_{sgx}) to obtain SK_{sgx}, then obtains the ciphertext data c_i and its corresponding hash value h_i through the smart contract, regenerates the ciphertext hash value h, and calculates whether h_i and h are equal to verify data integrity.

(2) The SGX corresponding to the query node uses the private key SK_{sgx} to decrypt $\text{Enc}_{PK_{sgx}}(k)$ to obtain the key $k = \text{Dec}_{SK_{sgx}}(\text{Enc}_{PK_{sgx}}(k))$.

(3) The SGX corresponding to the query node uses the key k to decrypt the ciphertext data c_i to obtain the plaintext data m_i for performing the search task.

Input: Session key k; A's data $M = \{m_1, m_2, \ldots, m_n\}$; Search P;
Output: Transaction txdata;
(1) Encrypted Owner data $M = \{m_1, m_2, \ldots, m_n\} c_i = \mathrm{Enc}_k(m_i)$ and $C = \{c_1, c_2, \ldots, c_n\}$;
(2) Encrypted kEkey $= \mathrm{Enc}_{\mathrm{PK}_{sgx}}(k)$;
(3) Set $h_i = \mathrm{hash}(c_i)$ and $H = \{h_1, h_2, \ldots, h_n\}$;
(4) Set $r = \mathrm{sign}_{\mathrm{SK}_o}(\mathrm{PK}_o \| H \| P)$;
(5) Set Set $= (\mathrm{PK}_o \| C \| H \| P \| r)$;
(6) Send Set to the DS and get addr;
(7) Generate timestamp $ts1$;
(8) Set txdata $= \{E$key, H, addr, $ts1\}$;
(9) Return txdata;

Algorithm 1: Privacy protection of data owner A's data.

(4) The SGX corresponding to the query node uses the key k to decrypt $\mathrm{Tr} = \mathrm{Enc}_k(w)$ to obtain the query keyword $w = \mathrm{Dec}_k(Tr)$.
(5) The SGX corresponding to the query node performs a search task on the plaintext data m_i according to the query keyword w. If the query is successful, the count value is incremented by 1.
(6) If count is not equal to 0, use the public key PK_u of the data requester B to encrypt m_i to generate $E\mathrm{data}_i = \mathrm{Enc}_{\mathrm{PK}_u}(m_i)$; else return false.
(7) Finally, the query node sends the encrypted data $E\mathrm{data}_i$ or false to the data requester B through the blockchain.

Input: $\mathrm{Enc}_k(w)$; $\mathrm{Enc}_{\mathrm{PK}_{sgx}}(k)$; c_i, h_i; count;
Output: search result;
(1) Set $h = \mathrm{hash}(c_i)$;
(2) if $(h = h_i)$ then
(3) Decrypt $\mathrm{Enc}_{\mathrm{PK}_{sgx}}(k) k = \mathrm{Dec}_{\mathrm{SK}_{sgx}}(\mathrm{Enc}_{PK_{sgx}}(k))$;
(4) Decrypt $\mathrm{Enc}_k(w) w = \mathrm{Dec}_k(\mathrm{Enc}_k(w))$;
(5) Decrypt $c_i m_i = \mathrm{Dec}_k(c_i)$;
(6) Search computation with m_i;
(7) if (count ! = 0) then
(8) Encrypted $m_i E\mathrm{data}_i = \mathrm{Enc}_{\mathrm{PK}_u}(m_i)$;
(9) return $E\mathrm{data}_i$;
(10) else
(11) return false;

Algorithm 2: Data acquisition and search computation.

5.1.6. Data Decryption Phase. Data requester B obtains $E\mathrm{data}_i$ from the blockchain and decrypts $E\mathrm{data}_i$ using the private key SK_u to obtain the corresponding plaintext data.

5.2. Our Protocol. In this scheme, the protocol flow includes the following 11 steps, where Step 1 describes the data preprocessing phase, Steps 2–3 describe the data storage phase, Steps 4–5 describe the user authorization phase, Steps 6–8 describe the data search phase, and Step 9 describes the data decryption phase. The logical process is shown in Figure 4.

Step 1. The data owner A encrypts data with an encryption key k and uploads it to the data storage DS.

Step 2. Data storage DS receives the encrypted data. If the verification passes, it returns the corresponding hash address; otherwise Step 1 needs to be performed again.

Step 3. Data owner A records hash address and encrypted key on the blockchain in the form of transactions for data sharing.

Step 4. If data requester B wants to search the encrypted data of data owner A, data requester B needs to send an authorization request to data owner A.

Step 5. Data owner A performs authorization verification and updates the legal user list after the verification is passed and uploads the Trapdoor to the blockchain.

Step 6. The query node obtains the Trapdoor, hash address, and encrypted key from the blockchain and then downloads encrypted data from the data storage DS.

Step 7. The query node performs decryption and keyword search operations of ciphertext in the SGX-based TEE.

Step 8. The query node sends the encrypted search results to the blockchain.

Step 9. Data requester B obtains the encrypted search results from the blockchain and then decrypts the search results.

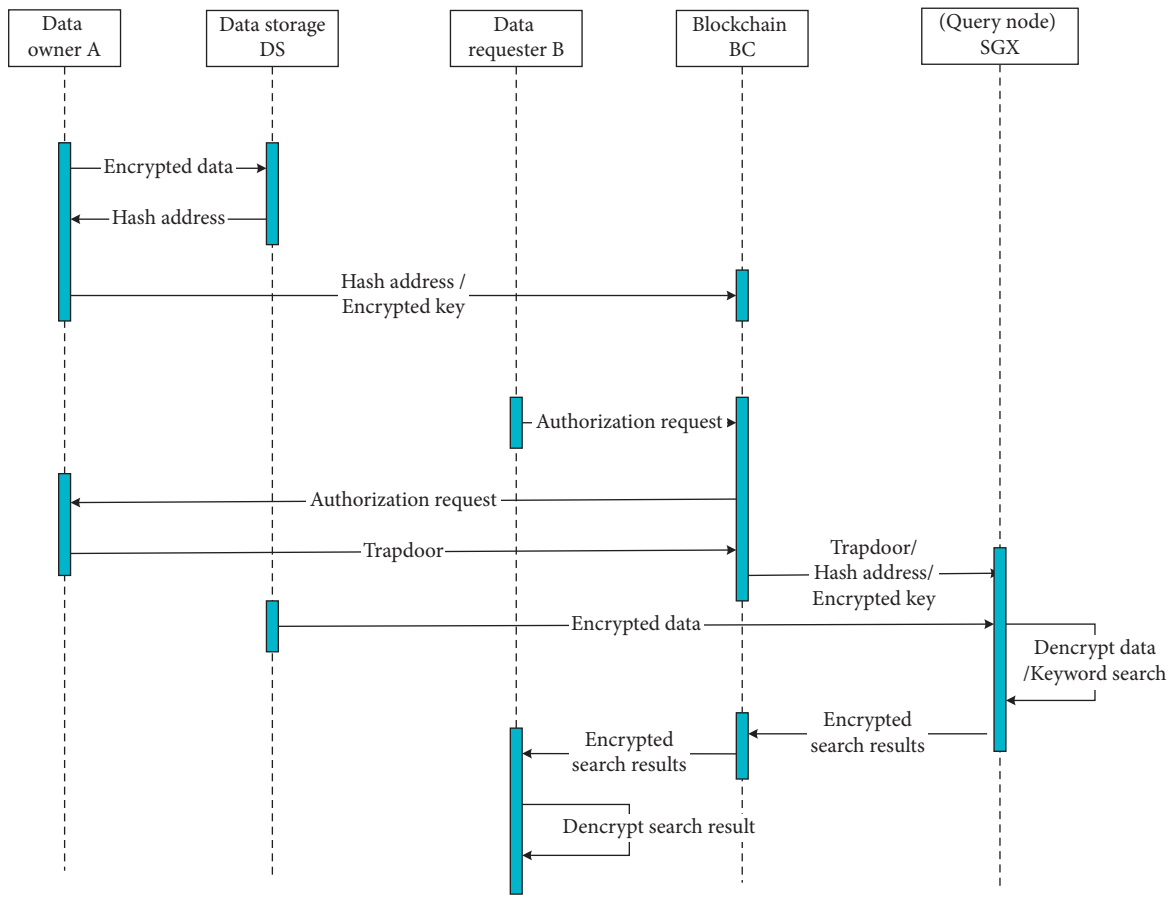


Figure 4: Logic process.

6. Security and Function Analysis

6.1. Security Analysis. Since the personal data of the data owner is sensitive and private, the security of the data is of great importance in this scheme. Data security is analyzed in the following aspects.

6.1.1. Data Security. In this solution, to protect data security, the personal data of data owner A is encrypted by the key k and stored in the data storage DS, and data requester B can obtain the search permission only through an authorization request. First, the data owner A uses the key to encrypt the data and store it in the data storage DS. Anyone can find the encrypted data through the storage hash address in the blockchain. To decrypt the encrypted data, the attacker must obtain the key k. However, the key k is only stored locally in data owner A and SGX security zone. Assuming that data owner A does not leak the private key, the attacker cannot obtain the key k. Therefore, the security of the personal data of data owner A is guaranteed.

6.1.2. Signature Forgery. In this scheme, the correct storage of data is guaranteed through the basic principle of signature. The user signs the ciphertext data and uploads it to the data storage DS. When the private key of the data owner is securely stored, the attacker cannot forge the signature, so other entities cannot destroy the authenticity of the data upload by forging the signature.

6.1.3. Tamper-Proof. In the data upload storage phase, there may be malicious users tampering with the blockchain information or transaction information. In this plan, the blockchain setup is POA consensus. Each block is generated by the certification node. For the compulsory process to verify the identity, the right to generate new blocks can be obtained. Malicious nodes cannot know the private key of the credible certification nodes, and it is impossible to fake the identity of the certification node to pack the block or modify the signature of the block information. Malicious nodes are difficult to tamper with data on the blockchain and data stored on the blockchain to ensure the authenticity and accuracy of the data.

6.1.4. Data Privacy. In this solution, to ensure the privacy of the data, data owner A uses a symmetric encryption algorithm to protect personal data, and data requester B can obtain the search permission only after the authorization is successful. In addition, in the data search stage, the encrypted personal data and query keywords are read in the nonsecure area of SGX, encrypted data can only be decrypted in the secure area of SGX, and the decrypted data cannot be obtained in the nonsecure area of SGX. Finally,

Table 2: The comparison of functionality and security with the existing schemes.

Schemes	Encryption algorithm	Pregenerated indexes	Authorized access	Blockchain-based	Multiuser scenario	Execution environment/leakage risk
Wang and Fan [29]	Symmetric	√	×	×	×	Cloud server/high
Chen et al. [30]	Public key	√	×	√	×	Smart contract/high
Liu et al. [10]	Symmetric	√	×	√	√	Smart contract/high
Fu et al. [15]	Homomorphic	√	×	×	√	Cloud server/high
Our scheme	Symmetric	×	√	√	√	SGX/low

use the public key of data requester B to encrypt the search results in the secure area of SGX and output them in the nonsecure area of SGX. Even if other machines steal the encrypted data in the nonsecure area, the data cannot be decrypted on the personal machine. Thus, the privacy and security of the data in the data upload stage and the data search phase are guaranteed.

6.2. Function Analysis. In Table 2, we compare our scheme with existing schemes in terms of functionality and security. We can see that schemes [10, 15, 29, 30] require pregenerated indexes and do not better solve the authorized access problem. Schemes [10, 30] use smart contracts to replace cloud servers to perform search tasks, which can solve the centralization problem of cloud servers in schemes [15, 29], but schemes [10, 30] still have a high risk of data leakage and are difficult to apply to complex computing scenarios problems. Also, schemes [29, 30] are suitable for single-user scenarios. However, this scheme does not require pregenerated indexes and is suitable for multiuser scenarios. Our scheme also uses TEE to better protect user data security and perform decryption and full-text search under the SGX off-chain to solve the problem that smart contracts cannot be applied to complex computing scenarios.

7. Experiment and Analysis

In this subsection, the practicality and feasibility of this scheme will be tested and analyzed through experiments. We installed Ubuntu 20.0 on a computer with Intel(R) Core(TM) i7-9750H CPU@2.60 GHz, 16 GB RAM, Microsoft Windows 10 operating system, and then performed simulation experiments. We use the DES symmetric encryption algorithm to encrypt the data, the blockchain part uses the Ethereum private network built by Geth, the smart contract is written in solidity language, and the search program P is written in C++. Next, we mainly tested and analyzed the implementation and gas costs of the contract and the performance of this scheme.

7.1. Implementation and Gas Costs. To discuss the feasibility of smart contract in this scheme, we implemented it on Rokeby (an Ethereum test network) where Rinkeby not only provides free funding requests but also designs a user interface for a convenient block resource manager. In addition, we employed a Google Chrome plugin (MetaMask-Chrome) to link Rinkeby in Chrome and use Remix10 to deploy and invoke smart contracts; the details of this implementation are shown below.

(1) First, we used MetaMask to generate two users (data owner A and data requester B) for our test with addresses 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 and 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2, then switched to A's account, and requested 3 Ether from Rinkeby so that A could deploy contracts and generate data transactions, etc.

(2) Then, we simulate user A. We use Rinkeby to deploy the smart contract to the blockchain and get its address (0x9D7f74d0C41E726EC95884E0e97Fa6129e3b5E99), and we also call the autuser algorithm via Remix to verify that B's authorization is passed or not.

(3) Next, we simulated A updating and viewing the list of legitimate users. Update the authorized user B to Rinkeby by calling the adduser algorithm, and view the list of legal users by calling the getuser algorithm. Here getuser is designed as a view type algorithm that will not modify the state of the smart contract (therefore, there is no transaction confirmation time).

(4) Finally, we also simulated that A deletes the authorized user B from the list of legitimate users. The authorized user B is removed from the legal user list by calling the deluser algorithm, where the algorithm can only be called by A.

Additionally, to test the cost in terms of transaction fees, we evaluated the gas cost of these operations (i.e., authorize, adduser, getuser, deluser). As it can be seen from Table 3, the biggest cost is to deploy a smart contract, which is about 5.0881 USD, but it only needs to be executed once. Although other operations are called repeatedly, their cost is about 1 USD (especially the cost of getuser is about 0.1709 USD), which means that A only needs to spend 0.1709 USD to view the list of legitimate users, which is an acceptable cost even if it is called repeatedly.

7.2. Performance Test and Analysis

7.2.1. Cost of Cryptographic Primitives. We use symmetric encryption algorithm to perform user encryption and SGX decryption tests on file data, respectively, and compare the

Table 3: Gas costs for smart contract (Gasprice = 2 GWEI, 1 ether = 2900 USD).

Operation	Gas used	Actual cost (ether)	USD
Deploy	876489	0.00175452978	5.0881
Autuser	34005	0.000068010	0.1972
Adduser	50096	0.000101920	0.3167
Getuser	29474	0.000058948	0.1709
Deluser	43749	0.000087498	0.2537

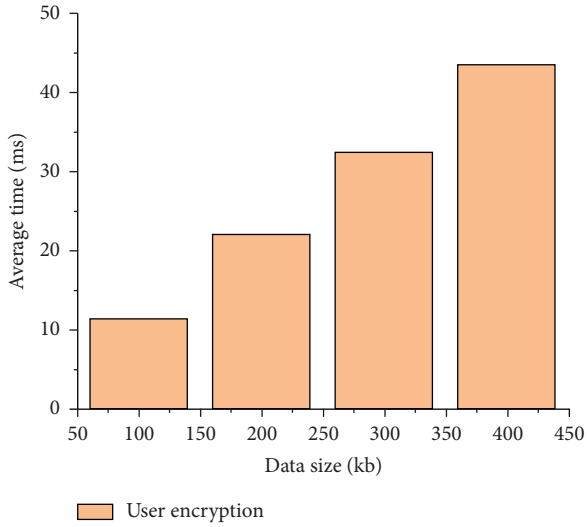


Figure 5: User encryption time test.

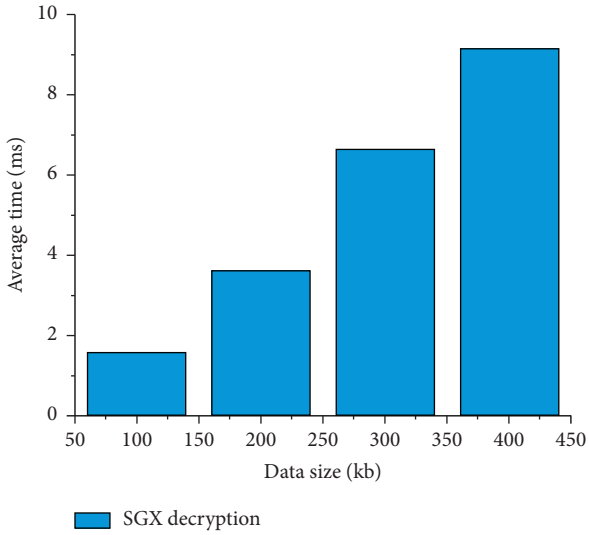


Figure 6: SGX decryption time test.

average calculation time of different sizes of data. Each experiment is repeated 1000 times and the average time is calculated. The test results are shown in Figures 5 and 6. As can be seen from the figures, the average time of user encryption and SGX decryption increases gradually with the increase of data volume.

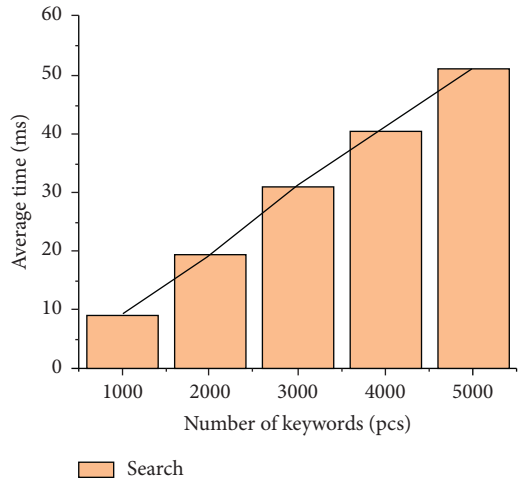


Figure 7: Search time test.

7.2.2. Search Performance. To verify the performance of this scheme in terms of search efficiency, as shown in Figure 7, we use the control variable method and the length of the keyword as 2 characters and test the average time of different numbers of keywords. The unit of time is milliseconds (ms), each experiment is repeated 1000 times, and the average time is calculated. From the curve shown in Figure 7, it can be seen that, with the increase in the number of keywords, the search time increases; this scheme only designs a simple keyword matching operation in the data search phase, which is related to the number of keywords. For encrypted data stored in the cloud environment, the search time efficiency of this scheme is reasonable.

8. Conclusion

In this paper, we propose a multiuser ciphertext search scheme which uses blockchain and SGX to protect keywords and data privacy and performs the decryption and keywords matching of ciphertext data in SGX. To fit searching in multiuser scenarios, we design a smart contract to realize user authorization and management. The security analysis, function comparison, and performance analysis prove that our scheme meets the security and privacy requirements. In the future, we will carry out further research work on multikeyword ciphertext search based on blockchain and SGX in specific scenarios.

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 work was supported by Shandong Provincial Key Research and Development Program (2021CXGC010107 and 2020CXGC010107) and the National Natural Science Foundation of China (62102209).

References

[1] D. 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, Berkeley, CA, USA, May 2000.

[2] R. A. Andreev, P. A. Andreeva, L. N. Krotov, and E. L. Krotova, "Review of blockchain technology: types of blockchain and their application," *Intellekt Sist Proizv*, vol. 16, no. 1, pp. 11–14, 2018.

[3] S. Jiang, J. Cao J, J. A. McCann et al., "Privacy-preserving and efficient multi-keyword search over encrypted data on blockchain," in *Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 405–410, IEEE, Atlanta, GA, USA, July 2019.

[4] X. D. Yang, T. Li, R. Liu, and M. Wang, "Blockchain-based Secure and Searchable EHR Sharing Scheme," in *Proceedings of the International Conference On Mechanical, Control And Computer Engineering (ICMCCE) 24th*, pp. 822–825, Hohhot, China, October 2019.

[5] Y. Zhang, J. F. Wang, and R. W. Wang, "Decentralized searchable encryption scheme based on dynamic accumulator," *Chinese Joural of Network and Information Security*, vol. 5, no. 2, pp. 23–29, 2019.

[6] Y. Guo, C. Zhang, and X. Jia, "Verifiable and Forward-Secure Encrypted Search Using Blockchain Techniques," in *Proceedings of the ICC 2020-2020 IEEE International Conference On Communications (ICC)*, pp. 1–7, IEEE, Dublin, Ireland, June 2020.

[7] M. Poongodi, M. Hamdi, V. Varadarajan, B. Rawal, and M. Ma, "Building an authentic and ethical keyword search by applying decentralised (blockchain) verification," in *Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 746–753, Toronto, ON, Canada, July 2020.

[8] G. Xu, S. Y. Xu, Y. B. Cao 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.

[9] S. Fu, C. Zhang, and W. Ao, "Searchable encryption scheme for multiple cloud storage using double-layer blockchain," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 16, 2020.

[10] X. R. Liu, G. J. Wang, B. W. Yan, and J. Yu, "KCB-BC-SSE: a keyword complete binary tree searchable symmetric encryption scheme using blockchain," *Procedia Computer Science*, vol. 187, pp. 377–382, 2021.

[11] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, pp. 149–180, 1978.

[12] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, pp. 169–178, Maryland: Bethesda, May 2009.

[13] M. V. Dijk, C. Gentry, S. Halevi, and V. Vinod, "Fully homomorphic encryption over the integers," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques 29th*, French Riviera, May 2010.

[14] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based," *Annual International Cryptology Conference*, vol. 8042, pp. 75–92, 2013.

[15] W. Fu, M. Li, and H. Zhao, "CRSHE: a novel ciphertext retrieval scheme based on homomorphic encryption," *Computer Engineering and Science*, vol. 40, no. 9, pp. 1540–1545, 2018.

[16] B. Han, Z. Li, and Y. Tang, "Design and implementation of full text retrieval scheme based on homomorphic encryption," *Computer Engineering and Applications*, vol. 56, no. 21, pp. 103–107, 2020.

[17] J. S. Liu, X. Wang, and H. Wang, "Keywords retrieval based on homomorphic encryption in cloud server," *Science Technology and Engineering*, vol. 21, no. 8, pp. 3180–3185, 2021.

[18] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M. C. Roşu, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries," *Advances in Cryptology*, vol. 8042, pp. 353–373, 2013.

[19] S. Jarecki, C. Jutla, H. Krawczyk, R. Marcel, and M. Steiner, "Outsourced symmetric private information retrieval," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, Berlin Germany, November 2013.

[20] D. Boneh, G. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proceedings of the Eurocrypt 2004*, pp. 506–522, Interlaken, Switzerland, May 2004.

[21] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," *Applied Cryptography and Network Security*, vol. 3089, pp. 31–45, 2004.

[22] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted Data," in *Proceedings of the International Conference on Information and Communications Security 7th*, pp. 414–426, Beijing, China, December 2005.

[23] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," in *Proceedings of the 2011 IEEE INFOCOM*, Shanghai, China, April 2011.

[24] H. Hu, J. Xu, C. Ren, and B. Choi, "Processing private queries over untrusted data cloud through privacy homomorphism," in *Proceedings of the IEEE International Conference on Data Engineering 27th*, pp. 601–612, Hannover, Germany, April 2011.

[25] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[26] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[27] H. Li, Q. Huang, J. Shen, G. Yang, and W. Susilo, "Designated-server identity-based authenticated encryption with keyword search for encrypted emails," *Information Sciences*, vol. 481, no. 8, pp. 330–343, 2019.

[28] Y. Yang and J. Liu, "Fast multi-keyword semantic ranked search in cloud computing," *Chinese Journal of Computers*, vol. 41, no. 6, pp. 1127–1139, 2018.

[29] B. Wang and X. Fan, "Lightweight verification for searchable encryption," in *Proceedings of the IEEE 17th International Conference On Trust, Security And Privacy In Computing And Communications*, pp. 932–937, New York, NY, USA, August 2018.

[30] 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.

[31] 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.

[32] Y. Li, L. L. Xu, W. H. Li et al., "Attribute Set-Based Boolean Keyword Search over Encrypted Personal Health Records," *Security and Communication Networks*, vol. 2021, Article ID 9023141, 13 pages, 2021.

[33] P. Zhang, Z. Chen, K. Liang, S. Wang, and T. Wang, "A cloud-based access control scheme with user revocation and attribute update," *Information Security and Privacy*, vol. 9722, pp. 525–540, 2016.

[34] X. Mao, J. Lai, Q. Mei, K. Chen, and J. Weng, "Generic and efficient constructions of attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 5, pp. 533–546, 2016.

[35] 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.

[36] Y. B. Miao, J. F. Ma, X. M. Liu, X. Li, J. Qi, and J. Zhang, "Attribute-based keyword search over hierarchical data in cloud computing," *IEEE Transactions on Parallel*, vol. 19, no. 6, pp. 985–998, 2017.

[37] W. C. Zheng, A. Wu, Y. F. Li, Q. Xing, and S. Geng, "Blockchain-Enabled Public Key Encryption with Multi-Keyword Search in Cloud Computing," *Security and Communication Networks*, vol. 2021, Article ID 6619689, 11 pages, 2021.

[38] H. Li, C. Gu, Y. Chen, and W. Li, "An efficient, secure and reliable search scheme for dynamic updates with blockchain," in *Proceedings of the 2019 the 9th International Conference on Communication and Network Security*, pp. 51–57, Chongqing, China, November 2019.

[39] X. Q. Wang, G. X. Cheng, and Y. Xie, "Efficient verifiable key-aggregate keyword searchable encryption for data sharing in outsourcing storage," *IEEE Access*, vol. 8, pp. 11732–11742, 2020.

[40] W. Zheng, Y. Wu, X. X. Wu et al., "A survey of Intel SGX and its applications," *Frontiers of Computer Science*, vol. 15, no. 3, Article ID 153808, 2021.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 2330052, 19 pages
https://doi.org/10.1155/2022/2330052

Research Article

Lightweight Mutual Authentication Scheme Enabled by Stateless Blockchain for UAV Networks

Lingjun Kong,[1] **Bing Chen**,[1] **Feng Hu**,[1] **and Ji Zhang**[2]

[1]*University of Southern Queensland, Toowoomba, Australia*
[2]*Nanjing University of Aeronautics and Astronautics, Nanjing, China*

Correspondence should be addressed to Bing Chen; cb_china@nuaa.edu.cn

Received 25 April 2022; Revised 22 June 2022; Accepted 28 July 2022; Published 14 September 2022

Academic Editor: Yinbin Miao

The UAV network composed of resource-constrained lightweight UAV swarms can efficiently accomplish mission with time critical requirements in dynamic and complex environments. However, the trusted authentication of network nodes poses a huge challenge due to its own resource constraints, the lack of trusted centralized support, frequent joining or departure of UAVs to or from the network, and the presence of cyber-attacks. In this paper, we propose a stateless blockchain based on triple aggregatable subvector commitment and present a dynamic proof of trust authorization consensus mechanism with a periodic random selection of authorized nodes to guarantee the trustworthiness of mutual authentication of UAV nodes. Our proposed triple vector authentication solution solves several of the challenges mentioned above very well. The extensive experiments demonstrate that our blockchain-based authentication scheme enjoins significant advantages over the four schemes currently available for UAV network authentication in terms of single authentication latency, speed of energy consumption, average computational cost, and end-to-end latency.

1. Introduction

The UAV network is a mission-oriented, temporary mobile self-organizing network, consisting of a fleet of lightweight UAVs that collaborate with each other at low cost; with distributed, equal, and destruction-resistant characteristics, all drones are linked as peer entities, both as data processing hosts and to undertake message routing and forwarding functions, interdrone communication without base station forwarding, to complete data transmission in a multi-hop manner, capable of complex environments, and high timeliness. It has a wide range of practical applications, such as joint search and rescue, environmental surveys, emergency communications, and military missions. Lightweight UAV nodes have the advantage of efficient networking and easy deployment, but at the cost of limited resources in terms of energy supply, storage, and computing power, which makes UAV networks a special type of mobile self-organized networks and face more complex network threats than MANETs [1, 2].

Firstly, the use of wireless links makes the UAV network more vulnerable to attacks launched from the links, which can come from all directions, and any node can be targeted. Ways of compromise include revealing secret information, jamming information, and impersonating nodes. Each node therefore needs to be in direct or indirect contact with the adversary. Further, the autonomy of nodes in UAV networks, operating in an unpredictable environment, increases the risk of nodes being captured, compromised, and hijacked, and thus in addition to being subject to external attacks, attacks launched from within by compromised nodes are more difficult to detect and more dangerous. Therefore, the operation of any node must adhere to a certain pattern rather than immediately trusting its peers. Finally, the mobility of UAVs, complex mission environments, and mission needs all make UAVs frequent access to the network, resulting in dynamic changes in UAV network topology and size, leading to a network with no clear defensive boundaries and statically configured security solutions that are not applicable. At the same time, invalid

network node information leads to increased end-to-end latency and higher routing costs, increasing the number of mutual communication failures and reducing the overall performance of the network.

In conclusion, mission UAV networks in complex and unknown environments are inherently very vulnerable and dynamic, and such characteristics bring new challenges to their security defense. It is necessary to build a lightweight and trusted global trust platform on UAV networks to achieve efficient authentication and key management to secure UAV networks, while also meeting the requirements of real-time, robustness, and dynamic adaptability of ad hoc mission networks.

As a special mobile self-organizing network, the nodes of the UAV network are mainly authenticated based on the threshold secret sharing technology authentication mode, certificate chain authentication, blockchain-based authentication mode, and stateless blockchain based on the cryptographic accumulator method, but due to the limited resources of the UAV network, the dynamic nature of these methods are not good enough to meet the needs in terms of computing, bandwidth, storage, and energy supply.

In the stateless authentication blockchain recommended in this paper, UAV nodes establish the local trust degree of neighboring nodes by monitoring each other's forwarding behavior with neighboring nodes. The network periodically performs data consensus on the local trust degree of the authorized agent node group and completes a decision consensus based on this; i.e., it counts the global trust degree of nodes, elects a new round of authorized agent groups, and resets the three-vector commitment weights. A new block is created with the decision consensus result, and the UAV blockchain network system is updated. Through the identity vector commitment in the new block, untrustworthy nodes are identified and isolated from the network, maximizing the availability and trustworthiness of the network nodes actually involved in the mission and enabling a new round of UAV identity authentication. The decision consensus result is stored in the blockchain, while local trust transactions as data consensus can be discarded after the decision consensus is reached and do not need to be on the chain, so the identity blockchain for UAVs is stateless and lightweight for fast authentication of inter-UAV communication.

The main contributions of this work are as follows:

(i) First, we introduce the new concept of triple vector commitment stateless blockchain in UAV networks. Using an aggregatable subvector commitment technology, the blockchain only records the dynamic changes of identity commitments in triple vectors instead of every authentication transaction. This not only enables lightweight blockchain storage, but also avoids the massive amount of recalculation in individual vector commitment due to membership changes. It greatly reduces the computational and communication overhead incurred by UAVs frequently entering and leaving the network and the isolation of untrustworthy nodes.

(ii) Second, we propose a novel dynamic multicenter trust authorization proof consensus mechanism, where a set of agent nodes are periodically elected as a blockchain consensus committee among all UAVs that have been registered to the mission network. The committee members are randomly and dynamically replaced periodically to sense the UAV flight dynamics in real time and monitor the nodes' reports on the abnormal forwarding behavior of their own neighboring nodes. New block generation and consensus are either achieved periodically or triggered to complete in time according to node identity status changes. This not only ensures consensus efficiency, but also significantly reduces the risk of blockchain consensus master nodes being tracked and locked, and improves the security of the consensus process.

(iii) Third, we propose the method of local mutual authentication of blockchain nodes. In each period of the blockchain, any node of the UAV network is a peer-to-peer full node. The UAVs only need to provide their own commitment witness to achieve localized two-way authentication which only involves giving the existence of vector commitment instead of traversing the whole blockchain. This reduces both the computational and communication complexities of UAV mutual authentication to a constant level.

(iv) We compare our scheme with several major existing MANET node authentication schemes, including remote direct anonymous authentication, threshold key sharing authentication, certificate-coin authentication by blockchain token method, and blockchain authentication based on cryptographic accumulator. The extensive experimental results demonstrate that our proposed scheme outperforms other competitive schemes in terms of single-step authentication latency, energy consumption, authentication computational overhead, and end-to-end latency.

The rest of paper is organized as follows. The related work is discussed in Section 2. The system model, including the network model, the threat model, and the blockchain model, is elaborated in Section 3. Section 4 describes the design details of our proposed vector commitment-based lightweight authentication scheme for stateless blockchains. In Section 5, the safety certification and performance analysis on our proposed scheme are conducted. Simulation results and analysis are illustrated in Section 6. Finally, the conclusion is presented in Section 7.

2. Related Works

For the distributed, self-organized, and autonomous characteristics of self-organized networks, according to different application models, domestic and international research mainly includes the authentication model based on threshold secret sharing technology [3], certificate chain-

based authentication, and blockchain-based authentication model.

In [4], the UAV remotely connects to the control center via a 4G wireless network using direct anonymous attestation (DAA) for remote authentication. However, this method requires the support of a remote center and is not very scalable. Using the threshold secret sharing technique, [5] proposed a distributed certificate-based authentication model where the certificate is partitioned into n shares, a share is allocated to the node acting as a distributed certificate authority (D-CA), and t of these shares are collected at authentication time to reconstruct the certificate. In the scheme proposed by Yi and Krave [6], the node uses flooding to send a certificate request (CREQ) and the D-CA responds with a certificate reply (CREP) as a response. The successful collection of t copies of the certificate shares node, and the user reconstructs the complete certificate. A valid certificate indicates successful authentication. This approach increases the communication overhead of the network and does not protect against black hole attacks launched by resource-powered malicious nodes.

[7–9] proposed to apply identity-based public key cryptosystems to MANETs, introducing distributed cryptography to propose a fully distributed identity-based scheme, and each node performs the process of issuing and managing certificates and maintains a certificate repository. The nodes complete mutual authentication through the chain of authentication formed by the certificate repository. The advantage is that there is no need for a certification center to authorize the management of worker certificates, avoiding the risk of a single point of failure. But the introduction of private key generators (PKGs) caused key escrow problems and the risk of impersonation attacks. Certificates and identities cannot be bound, and malicious nodes can impersonate other nodes to join the network at will. In addition, the inconsistency of the certificate chain of each node also leads to authentication failure, and the certificate repository management and maintenance costs of the nodes increase with the expansion of the network scale. This is difficult to achieve for resource-constrained UAV nodes.

Certificate-less public key passwords [10] are an improvement on identity ID-based public key passwords, and [11–14] combined threshold cryptography with certificate-less public key passwords in the MANET authentication model. However, the security of the system master key relies on the absolute security and reliability of the distributed server, and in addition, there is a risk of man-in-the-middle attacks during key negotiation. Most of the schemes in the above literature use bilinear pairing, which provides good security, but their high operational complexity results in these schemes not being lightweight; key distribution mostly requires the establishment of a secure channel. Ad hoc, highly dynamic UAV networks cannot be provided.

Blockchain-based decentralized authentication uses the tamper-evident and traceable nature of the blockchain to store information such as identity and public key. The process of authentication traverses the blockchain to query the certificate, then checks whether the public key belongs to its declared identity, and finally sends a challenge message to determine whether the other party holds a matching private key by verifying the digital signature. [15] proposed authentication and key management mechanisms to achieve security of heterogeneous drones through the combination of transaction chain and blockchain, but the scheme requires that the drones as cluster head must have sufficient resources and act as the full node role of the blockchain, so there is still the risk of local single point of failure, which cannot guarantee the security of the full node of the cluster head itself, and the nonstop growth of the blockchain shared ledger makes the section face problems such as "storage bloat" and reduced authentication efficiency.

Researchers [16, 17] used blockchain technology to improve the public key infrastructure (PKI) authentication technology. Distributed PKI authentication is implemented to avoid the problems of single point of failure and certificate transparency in traditional PKI, and to effectively address the inefficiency of using the method of traversing the blockchain to query certificate authentication and the increasing storage overhead as the size of the blockchain grows. By combining blockchain and dynamic accumulator, a blockchain PKI model that can update certificates in bulk is constructed, thus improving the efficiency of authentication. The model can efficiently add, revoke, and renew user certificates. However, the consensus of the blockchain until the transaction is on the chain confirms that the authentication is successful, which makes the latency of a single authentication, as well as the computational and communication overheads insufficient to meet the requirements of mission drone networks in terms of real-time and low energy consumption. [18] Color green addressed this paradox by proposing a novel semipermitted blockchain framework that balances decentralization and efficiency, making the system scalable and efficient at the same time. A randomly selected public node joins the committee to execute the protocol to protect the block, but separates transaction execution from the protocol, thus reducing protocol waiting time and allowing lightweight nodes to participate, but the public node requires high resources.

The combination of blockchain technology and cryptographic accumulator technology has been used to solve the authentication problem of distributed network systems, and there have been many research results at home and abroad. The accumulator, first proposed by Benaloh and de Mare [19], is a compact representation of an arbitrarily large set that can be used to prove claims of membership or non-membership in the underlying set. The protocol in [20] used RSA accumulators to combine large states into a short commitment to design stateless blockchains where the verifier only needs to store block headers, greatly reducing the need for disk and RAM, reducing the storage overhead of the verifier, and linearly increasing system throughput. [21] provides cryptographic accumulator universal composable (UC) processing using two weaker accumulators, constructing the accumulator in a modular fashion and extending the anonymous credential system to support revocation using the results of the UC accumulator. Libert and Yung in [22–24] vector commitments give

Table 1: Classification and comparison of authentication methods.

Method	Papers	Overhead and shortcomings
Threshold secret sharing	[3, 5, 6]	High computational and communication overheads; unable to defend against black hole attacks launched by malicious nodes with powerful resources.
Certificate chain	[7–9]	High storage and communication overheads; there are key escrow issues and risk of impersonation attacks. Inconsistencies in the certificate chain across nodes lead to authentication failures. As the size of the network increases, the cost of managing and maintaining the certificate store increases.
Certificate-less public key	[10–13]	High computational and communication overheads; man-in-the-middle attack risk during key negotiation, key distribution mostly requires establishment of secure channels.
Traditional blockchain	[15–17]	High storage and computational overheads, "storage explosion," inefficient consensus, and limited system scale.
Stateless blockchain	[21–24]	Storage overhead very low; nodes are dynamically added and removed, resulting in frequent recalculations of the accumulator.

commitments to ordered sequences that satisfy positional binding; i.e., an adversary algorithm should not open a commitment for two different values at the same position. The commitment string and the open witness are short, and their size is independent of the vector length. [20] applies unknown-order group batch processing techniques to cryptographic accumulators and vector commitments to develop techniques for noninteractive aggregated membership proofs that are verified by a constant number of group operations and provide size invariant bulk nonmembership proofs for a large number of elements. Using these new accumulator and vector commitment constructs to design stateless blockchains where nodes require only a constant number of stores to participate in consensus. [25] proposed vector commitments with subvector openings that allow a commitment vector to be opened at a set of locations with an opening size that is independent of the length of the vector and the number of open locations. On its basis, [23] proposed incremental aggregation to design an algorithm that generates openings quickly by preprocessing and then to implement subvector commitments. VMware research and the Ethereum team [24] propose aggregatable subvector commitment (aSVC) schemes that can aggregate multiple proofs into a small subvector proof. The approach of aSVC obtaining a stateless payment cryptocurrency has very low communication and computational overhead. However, the above authentication methods complete consensus on a fixed number of nodes and all suffer from accumulator recalculation when nodes leave or join. The joining and leaving of drone nodes in a UAV network are frequent, and there is interference from Byzantine nodes with legitimate identities, which the above parties cannot handle. Table 1 summarizes the above authentication methods.

The authentication methods described above cannot be applied to lightweight, dynamic, and time-varying node trustworthiness for UAV networks. How to build a dynamic UAV trustworthy platform based on stateless blockchain to provide fast mutual authentication between UAVs is the main research objective of this paper.

3. System Models

UAV networks in complex and unknown mission environments are inherently Byzantine distributed systems with time-varying trustworthiness. The purpose of the lightweight authentication blockchain system is to monitor the trustworthiness of drone nodes during a mission and to provide a global platform for rapid mutual authentication between nodes. In traditional blockchains, transactions need to complete consensus and update the blockchain across the network before they can be authenticated successfully, which makes the authentication efficiency, and the computation and communication overhead insufficient to meet the requirements of UAV networks in terms of real-time and low energy consumption. The stateless authentication blockchain provided in this paper periodically performs data consensus on the local trustworthy state records of nodes, which are generated by monitoring the forwarding behavior of neighboring nodes, and then performs decision consensus on the data consensus results, i.e., aggregatable identity vector commitment based on the global trustworthiness of nodes. Its lightweight nature is reflected in the fact that only the decision consensus result is kept, and the new blocks added to the blockchain are blockheads of fixed size, without the need to keep intermediate historical state data; thus, its storage is controlled.

3.1. Network Model. In the mission preparation phase, the system authorizes the registration server as the authoritative control center in the initialization phase of the system, which initializes the security environment parameters of the mission. The UAV nodes and the created blocks of the blockchain register the UAV identity, calculate the identity vector commitment, and select the authoritative UAV node for the task execution phase. The proof-of-authority consensus mechanism (POA) is used to broadcast the created block to all the mission UAV nodes on the chain for reaching a consensus.

The system network model is divided into a network model for the mission preparation phase and a network model for the mission execution period based on the process of the mission (Figure 1). In the mission preparation phase, the UAV swarms and the registration server form a wireless network with the registration server as the authorization center in a secure environment. All nodes deploy blockchain client programs, and the registration server acts as a trusted authority to initialize the security environment parameters of the UAV mission network. The registration server acts as a

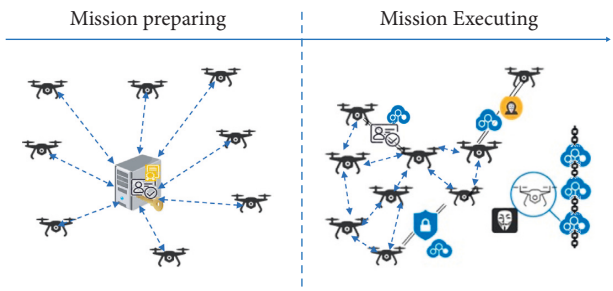


Figure 1: Mission-based UAV network model.

trusted authority to initialize the security environment parameters of the UAV mission network, register the identity of the UAV, assign public and private keys, establish the genesis block, and build the blockchain network system with the proof-of-authority consensus mechanism. The registration server does not participate in the mission execution, and the network after the mission starts is a self-organized network of autonomous UAV nodes that forward data in a multi-hop manner. The blockchain system supervises the flight dynamics and forwarding behavior of the network nodes in real time to maintain the effective operation of the mission network.

3.2. Threat Model. The ultimate goal of a mission-oriented UAV network is to complete time-sensitive missions, and any factor that affects the proper achievement of the mission can be considered a threat to the UAV network.

(i) Environmental threats: The UAV network mission execution environment is complex and variable, it may be the scene of distress and rescue, or it may be the enemy-occupied area of the battlefield, the UAV network may suffer physical interference, or even be directly damaged and affect the performance of the overall network, and the network system should have the ability to sense the nodes leaving the network in a timely manner and cancel the identity of the lost network members; at the same time, the additional network members can be quickly authenticated into the network. The network system should have the ability to sense when a node has left the network, to cancel the identity of lost network members, and to quickly authenticate additional network members to the network to ensure the network's ability to perform its mission.

(ii) Malicious nodes: Malicious nodes include external unauthorized malicious nodes and compromised nodes. Malicious nodes can launch impersonation attacks, black hole attacks, and DOS attacks, and can also conspire to conduct wormhole attacks. Compromised nodes with legitimate identities can be more damaging to the network by launching internal attacks. Therefore, in addition to authentication, the drone network should also have the ability to detect untrustworthy nodes and isolate compromised nodes from the network in a timely manner.

(iii) Selfish nodes: Due to their own reduced energy, nodes only receive information and do not forward it out of self-protection. Such uncooperative zombie nodes, although they do not initiate harmful attacks, exist in the network and generate ineffective communication, wasting energy and reducing the overall performance of the network. The system should also have the ability to identify and mark them for isolation.

3.3. Blockchain Model. The solution recommended in this paper implements local mutual authentication of UAV network nodes using a stateless authentication blockchain. The initialization of the blockchain is done in a secure environment. The mission starts with all UAV network nodes having the same Genesis block, which contains an identity vector commitment, an authenticated smart contract, and a specified set of authorized nodes. The consensus process takes place in the authorized node group, with the number of authorized nodes set based on the network size. The authorized nodes are responsible for detecting the flight status of the drone nodes, such as whether they leave the network. All nodes send to the authorized nodes the local trust assessment of neighboring nodes generated during the consensus cycle. Similar to the node trustworthiness monitoring method (WatchDog) proposed in [26], monitor the forwarding behavior of neighboring nodes to assess their trustworthiness. The consensus cycle is set according to the network size, but consensus is initiated when two conditions occur during the consensus cycle: (i) an authorized node finds a record below the trustworthiness threshold in the collected local trustworthiness assessment dataset; (ii) an authorized node does not receive a response from a particular drone node several times in a row, and this number exceeds the threshold set by the system.

The consensus process consists of a data consensus and a decision consensus. The data consensus consists of a local trustworthiness assessment generated by all nodes during the consensus cycle, and the status records of the UAV flights detected by the authorized nodes (whether they respond or not). Data consensus results in each authorized node having an identical subset of status records. A decision consensus is performed on the results of the data consensus to determine the global trustworthiness of the nodes, elect a new set of authorized nodes, and update the triple identity vector commitment. The results of the above decision consensus are recorded in a new block, a fixed size block header to be exact, and the drone network continues to work under the management of the new authorized node group after the blockchain has been synchronized and updated. In the meantime, historical state data used for data consensus can be discarded after decision consensus, and the blockchain grows only the block head that holds the decision consensus result at a time, avoiding the creation of a "storage explosion."

The consensus process is generated periodically, and the group of authorized nodes for consensus in each period is dynamically generated according to the consensus result of

the decision, which is a dynamic polycentric proof-of-authority consensus mechanism (DPOTA), as shown in Figure 2, where the UAV network is reorganized by new blocks added to the blockchain, triple identity vector commitment, node cancellation determined by dynamic aggregation, and isolation. The stateless blockchain UAV network guarantees network trustworthiness and provides fast mutual authentication between nodes.

4. Recommended Scheme

In this section, we explain the stateless blockchain authentication system supporting DPOTA consensus mechanism, and our approach solves or alleviates the conflict between UAV networks with resource constraints in storage, computation, energy, and bandwidth and high requirements in dynamism, real time, and security during mission execution. Figure 3 shows the operation of the mission-oriented UAV network stateless blockchain light authentication certification by timeline.

The mission-oriented stateless blockchain authentication system for UAV networks consists of two phases and four roles. In the mission preparation phase, the UAV network operates in a secure network environment, including a trusted third party, a registration server (RS), and a UAV to be registered (UAV); in the mission execution phase, it works in a nonsecure network state, including a stateless blockchain trusted platform and a blockchain UAV node (BUAV), and throughout the mission, the UAV network security is performed by the registration server and the blockchain together.

At the beginning of the mission, a blockchain client program is deployed for the registration server and the candidate UAVs participating in the mission to initialize the UAV network in a secure environment with the registration server as the center. The registration server constructs the UAV network mission-related security environment parameters based on the hyperelliptic curve public key cryptosystem [27] (HECC), receives UAV registration requests, generates public and private keys and identity IDs, and builds the identity vector. The vector commitment is calculated based on the identity vector, and the identity witness of the corresponding UAV is generated at the same time. Subsequently, trust authorization committee members are randomly selected, node trust vectors are initialized, and creation blocks are constructed. After completing the initialization, the registration server broadcasts the Genesis block to all registered UAVs to build the blockchain system of the UAV network.

4.1. System Initialization. In the mission preparation phase, the network environment is secure and the registration server is authorized as the control center to complete the initialization of the stateless blockchain system. The mission-oriented UAV network system is initialized, including the initialization of the registration server, the initialization of the UAV, and the initialization of the blockchain. Table 2 lists the main authentication-related global symbol.

Registration server initialization: First, the hyperelliptic curve $\mathrm{HE}(F_p)$ is customized for the system, where $p \in \mathrm{HE}(F_p)$ is its basis, the large prime q is its order, $q \neq p$, and q is not divisible by $p-1$. Then, set the one-way hash functions by equation (1), where $G_1 \subseteq (C, F_q)$ is the Abelian cyclic additive group on the hyperelliptic curve, generating the element $P \in G_1$.

$$\begin{aligned} H_1 &= (0,1)^* \longrightarrow Z_q^*, \\ H_2 &= (0,1)^* \longrightarrow G_1^*. \end{aligned} \tag{1}$$

Randomly select $k \in Z_q^*$ as the private key of the registration server and $P_k = kP$ as its public key. The public cryptographic parameters, $\{q, G_1, P, P_k, H_1, H_2\}$, are stored in the registration server only as important security environment parameters for the current mission.

UAV initialization: The UAV provides hardware-related information such as MAC and IP address, and applies for identity registration with $\{U \| U_{\mathrm{mac}} \| \mathrm{IP}\}$ as a request to the registration server, which is not involved in the mission execution. The registration server generates the private key $d \in Z_q^*$ and the corresponding public key $U = d \cdot P$ for the UAV. The public security parameters, $\{q, G_1, P, P_k, H_1, H_2\}$, are built into the associated smart contract in binary form, which is deployed to the Genesis block by the registration server. Based on the UAV identity request $\{U \| U_{\mathrm{mac}} \| \mathrm{IP}\}$, the registration server key $k \in Z_q^*$ is used to sign the requested UAV, and the registration smart contract generates the UAV node identity and assigns the initial value of trust to each node, with the identity ID calculated by equation (2). The final registration server assigns the public and private keys of the UAV, the identity ID, and the creation block to the corresponding UAV nodes.

$$\mathrm{ID}_i = H_2\left(H_1\left(U_i \| \mathrm{ID}_i \| \mathrm{IP}\right) \| \mathrm{Sign}_{\mathrm{reg}}^{k}\left(U_i \| \mathrm{ID}_i \| \mathrm{IP}\right)\right). \tag{2}$$

Stateless blockchain initialization: During the mission preparation phase, the network environment of the registration server is secure and the setup $(1^\lambda, 1^N)$ function is run to establish vector committed common reference parameters (crs), which are built into the smart contract associated with the creation of the block in binary form. Since the registration server does not participate in the task network, the crs of the UAV network are hidden during the mission execution phase and no adversary algorithm can use the crs to fake the related information. The structure of the Genesis block is shown in Figure 4, which mainly includes the registered UAV identity vector commitment, the consensus committee member list, the UAV trust value vector, and the smart contracts related to registration, deregistration, trust management, and authentication. The UAV identity registration contract (SC_IDReg) is invoked only at the registration server. The hyperelliptic curve cryptosystem is used to sign UAV requests and generate unique UAV ID. The order of UAV registration forms the order of positions in the identity vector, and values in the trust vector are assigned in this order. The number of registered UAVs can be much larger than the number of UAVs for mission execution.

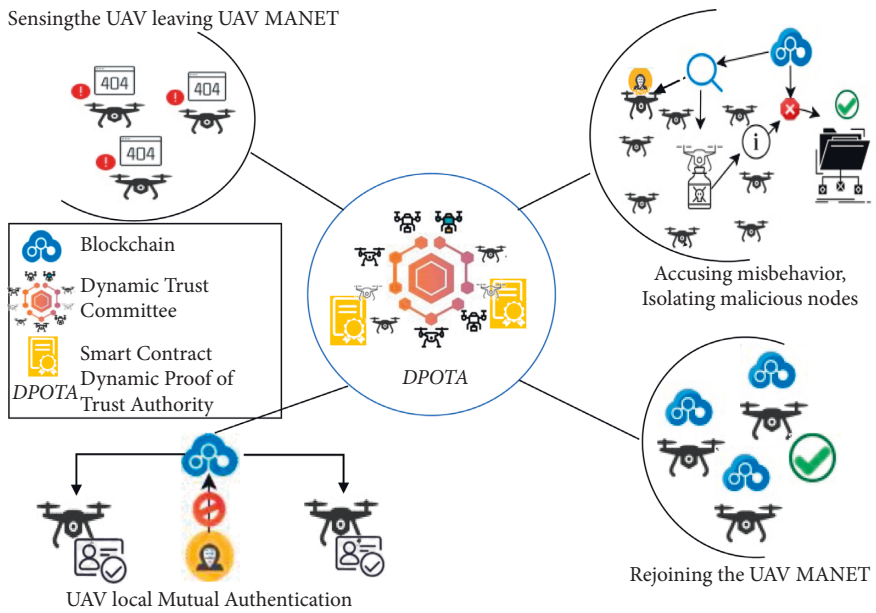


Figure 2: Stateless authentication blockchain model during mission execution.

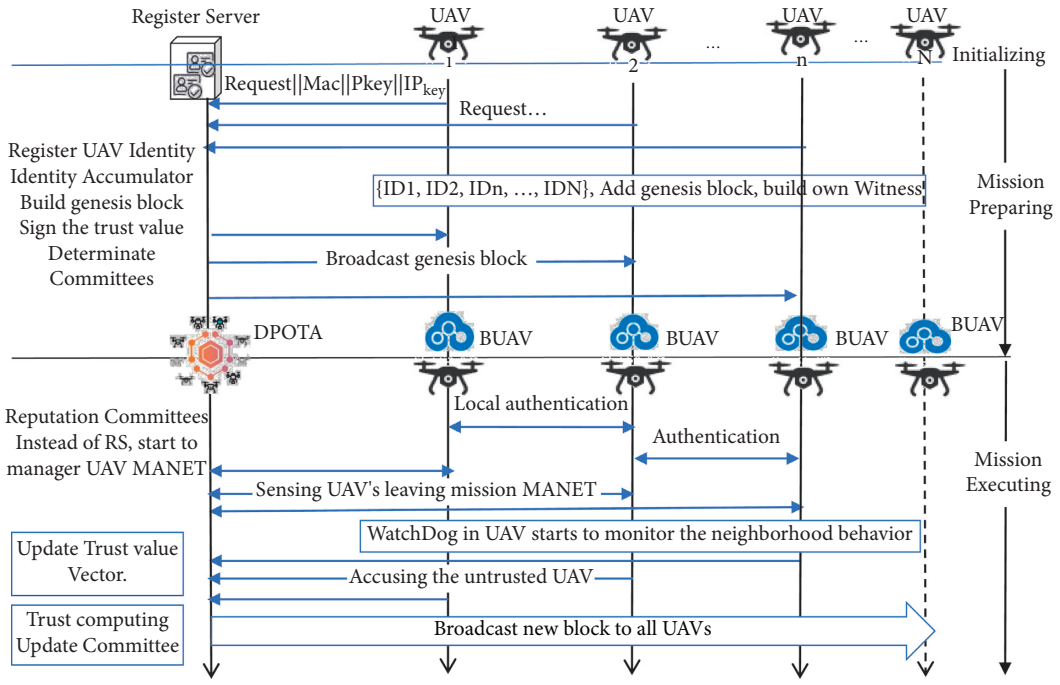


Figure 3: Mission-oriented UAV network blockchain workflow.

In the registration server, the smart contract, vector commitment accumulator (SC_VCCom), completes the registration of UAVs, generates identity witnesses, and builds vector commitments for all registered UAVs. After determining the UAVs to participate in the mission execution, t UAVs are randomly selected (t is set by the system in advance according to the application requirements) and their identity information key-value pairs, {ID: Pubkey, IPaddress}, are used to construct the initial list of trusted authorized members. These t UAVs are used as the blockchain consensus committee members in the first round of the mission execution phase.

Table 2: Global symbol.

Symbol	Description
ID_i	The i-th UAV identifier
W_i	The i-th UAV witness
C_{ID}	Identity vector commitment
$\widehat{W}$	Witness aggregation
$\widehat{W'}$	Aggregation of removed witnesses
G_1	The Abelian cyclic additive group of the hyperelliptic curve
$\overrightarrow{ID}$	UAV network node identity vector
$\overrightarrow{W}$	UAV network node witness vector

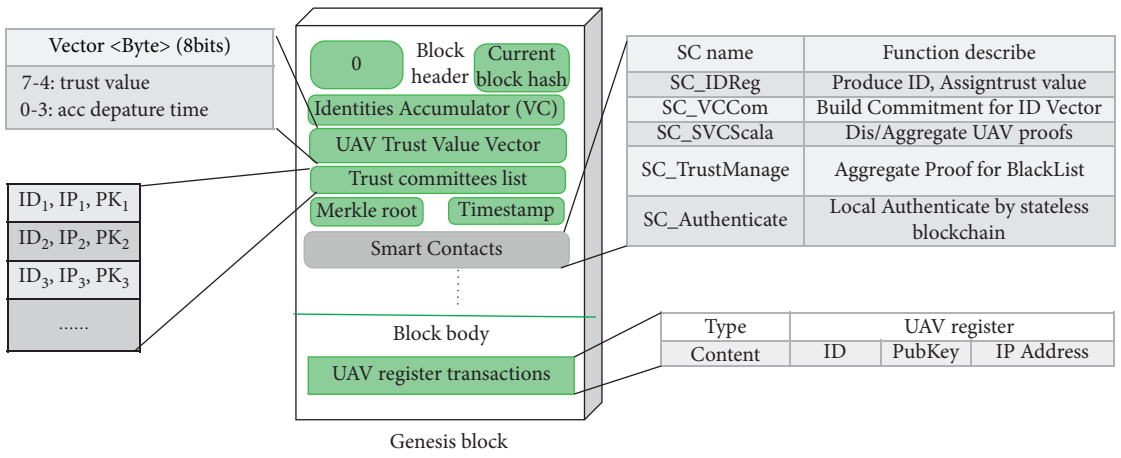


Figure 4: Stateless blockchain genesis block structure.

Input: Security parameters λ, UAV number N, UAVs request.
Output: Nodes' ID vector commitment, C_{ID} trust value vector.

In Register Server:
#Received all UAVs' requests
$\overrightarrow{\mathrm{ID}} = \{0\}$; TrustList = {0};
crs = Setup $(1^{\lambda}, 1^{N})$;
for i in N**do**# UAV nodes
  MySign = $\mathrm{Sign}^{k}_{\mathrm{reg}}(U_i \| \mathrm{MAC}_i \| \mathrm{IP})$
  $\mathrm{ID}_i = H_2(H_1(U_i \| \mathrm{MAC}_i \| \mathrm{IP}) \| \mathrm{MySign})$
  $\overrightarrow{\mathrm{ID}}$.Append$(\mathrm{ID}_i)$;
  TrustList.Append(trustvalue);
end for
#Get the all registered UAVs identities:
$\#\left\{\overrightarrow{\mathrm{ID}} = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_i, \ldots, \mathrm{ID}_N\}\right\}$
$C_{\mathrm{ID}} \longleftarrow$ Commit$(\overrightarrow{\mathrm{ID}}, r)$, #$r$ is randomnal;
#Randomly selects 5 UAVs from n UAVs as the trusted committee
IninilizeTrustList( )
for all UAV nodes:
for uav_i in length of $\overrightarrow{\mathrm{ID}}$. **do**
  $W_i \longleftarrow$ Prove$(i, \overrightarrow{\mathrm{ID}}, k)$, # k is randomnal;
  Send $(\overrightarrow{\mathrm{ID}}.\mathrm{ID}_i, W_i)$
end for
#When receive genesis_block from register sever
if current_block is constructed correctly **then**
  block_chain.append(genesis_block);
else
  Abort Genesis block;
end if

Algorithm 1: UAV registers/builds the stateless blockchain.

The identity vector is generated in the registration server $\overrightarrow{\mathrm{ID}} = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_i, \ldots, \mathrm{ID}_N\}, i \in \{1, 2, 3, \ldots, N\}$, combined with a random number to compute the identity vector commitment of the UAV, C_{ID}, and the identity witness vector $\overrightarrow{W} = \{W_1, W_2, \ldots, W_3, \ldots, W_N\}$. The registration server constructs the Genesis block and synchronizes it to all registered UAVs. The registered UAVs obtain their own IDs and identity witness to initialize the mission-oriented UAV network blockchain system. Please refer to Algorithm 1.

4.2. Triple Vector Commitment Stateless Blockchain. In the mission execution phase, the network environment is complex and insecure; with the possibility of external network attacks, nodes leaving the network, and nodes being compromised, the stateless blockchain serves as a global trust platform to manage the mission UAV network.

Dynamic multicenter proof-of-authority consensus protocol: When a new block is created, the current authoritative nodes randomly select the consensus committee members for the next round based on the blockchain trust

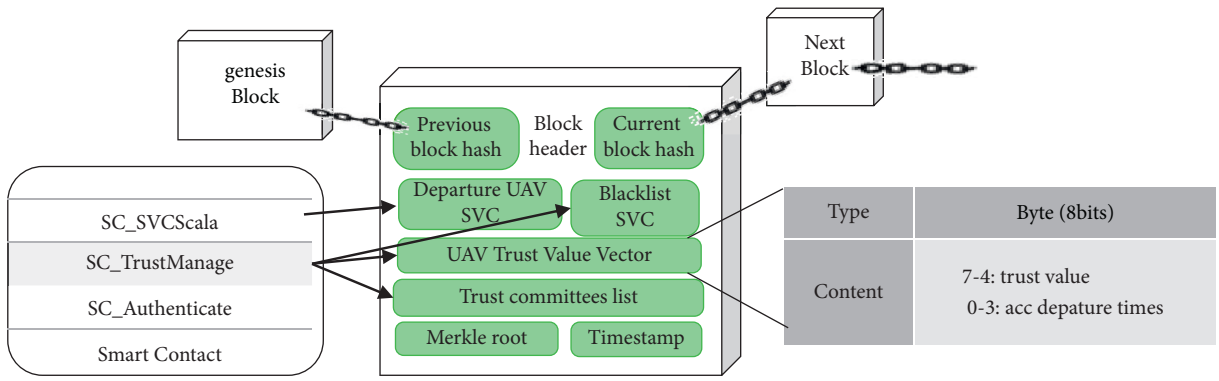


Figure 5: Stateless blockchain structure diagram.

vector. This makes it difficult for adversaries to ascertain the target to attack Through a smart contract related to trust management, the consensus committee members respond to the flight status of the drones and handle reports of abnormal behaviors when nodes forward data. The consensus mechanism is triggered directly when the aggregatable deregistration subvector or blacklist subvector of consensus nodes changes to ensure the trustworthiness and validity of participating members in the mission-oriented UAV network.

Figure 5 represents the structure of a new block added to the stateless blockchain, a fixed size block header that holds the results of each cycle of decision consensus, containing subvector witness aggregation, subvector witness aggregation for nodes leaving the network, subvector witness aggregation for untrustworthy nodes, and a dynamically changing vector of trust values for all nodes. The system sets the blockchain consensus period according to the network size and specific environment, and the historical state data used for data consensus need not be on the chain. Consensus committee members call the smart contract SC_SVCScala to perform dynamic aggregation of drone member witness and call the smart contract SC_TrustManage to modify the trust vector value of the drone. After the decision consensus, if any drone's trust value is below a certain threshold, its witness will be aggregated into the malicious node blacklist subvector; the witness of a drone that does not respond to the authorized node detection with a test greater than a set value will be aggregated into the revocation subvector. The number of authorized node groups is relatively small, and the PBFT consensus algorithm can be used for data consensus.

Identity vector commitment: Mission-oriented UAV networks operate in unknown and complex mission environments. The mission process is exposed to multiple risks, such as environmental factors causing nodes to leave the network, or compromise of internal nodes due to malicious attacks, and selfish behavior of nodes protecting their own resources. The UAV network needs to sense the dynamic changes in the validity and trustworthiness of UAV nodes in a timely manner. Rapid response to the deregistration, restoration, or isolation of abnormal nodes is necessary to maintain the overall performance of the network and ensure the reliability of mission execution. The proposed triple identity vector commitment mechanism avoids costly recalculation of the generic cryptographic accumulator due to changes in membership status and only requires reclassification of the changing UAV identity proofs. The key functions of the proposed scheme are shown below:

(1) $\text{crs} \leftarrow \text{Setup}(1^{\lambda}, 1^{N})$, output public parameter crs, supported vector length N, (crs include public parameters of the security environment of this mission network, providing implicit input for other algorithms, including adversary algorithms, and UAV network applications need hidden processing).

(2) $C_{\text{ID}} \leftarrow \text{Commit}(\overrightarrow{\text{ID}}, r)$, input vector $\overrightarrow{\text{ID}}$ and random number r, output vector of commitment C_{ID}.

(3) $\overrightarrow{W_i} \leftarrow \text{Prove}(i, \overrightarrow{\text{ID}}, r)$, generating witness of the existence of the corresponding element at position $i \in [N]$ in the $\overrightarrow{\text{ID}}$ vector.

(4) $\widehat{W} \leftarrow \text{Aggregate}(C_{\text{ID}}, \overrightarrow{\text{ID}}[S], \{W_i : i \in S\})$, given the set of positions $S \subset [N]$ of the elements of the vector to be aggregated, has been witnessed accordingly $W_i : i \in S$, and outputs aggregation $\widehat{W}$: $|\widehat{W}| = |W_i|$.

(5) $\widehat{W'} \leftarrow \text{Disaggregate}(\widehat{W}, \overrightarrow{\text{ID}}[S'], \{W_j : i \in S\}')$, unmake the corresponding witness in the set $S' \subset [N]$ from the aggregated $\widehat{W}$.

(6) $b \leftarrow \text{Verify}(C_{\text{ID}}, \overrightarrow{\text{ID}}[S], \widehat{W})$ verifies whether the commitment C_{ID} contains the corresponding subvector, $\overrightarrow{\text{ID}}[S]$, in the location set S by aggregating the witness $\widehat{W}$, and $b = 1$ indicates that the corresponding identity ID is legitimate.

In the mission preparation phase, the legal information of all nodes' ID witness is compressed into the identity vector commitment, and the UAV is assigned the identity ID_i in the registration phase, as well as the witness W_i that proves its existence in the commitment C. The first layer of vector commitment, $C_{\text{ID}} \leftarrow \text{Commit}(\overrightarrow{\text{ID}}, r)$, is created by the registration server and saved in the Genesis block. UAVs that become members of the consensus committee initiate the UAV flight state sensing module, which aggregates the identity witness of UAVs that have left the network to the revocation subvector commitment (the second layer vector commitment). During the mission execution phase, when the

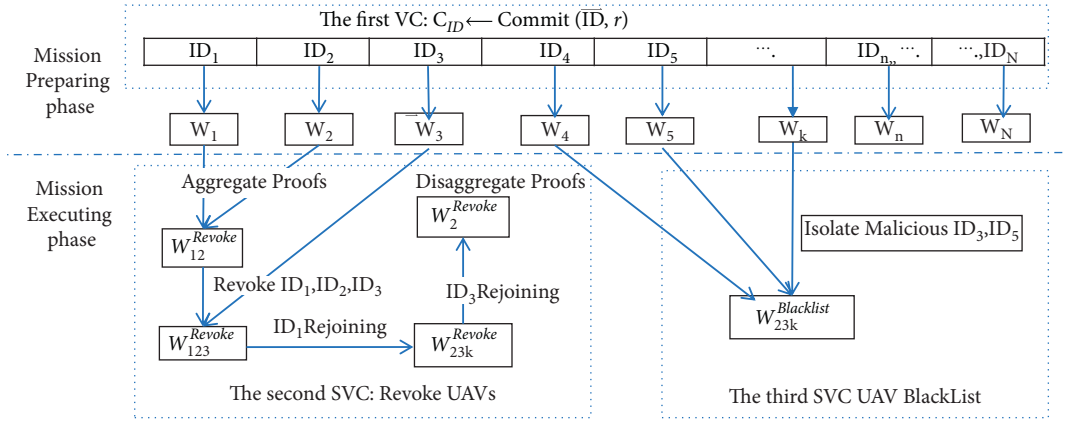


Figure 6: UAV network triple authentication vector commitment.

UAV forwards data, its built-in monitoring module WatchDog [12] reports the bad behaviors of neighboring nodes to authorized nodes. The smart contract related to trust management of the blockchain system determines whether to aggregate the identity witness of the questioned nodes to the blacklist subvector (the third layer vector commitment) based on their trustworthiness. As shown in Figure 6, when a UAV launches a communication request, the received UAV verifies whether it is in the identity vector commitment in turn, then detects whether its witness is in the blacklist subcommitment, otherwise detects whether its witness is in the revocation subvector commitment, and finally decides whether to de-aggregate the witness of the UAV from the revocation subvector, and de-aggregation means that the UAV rejoins the network. This ensures that the UAVs participating in the mission network are valid and trusted.

4.3. Identity Revocation Subvector Commitment. During the execution of the mission, the UAV leaves the network actively due to the mission need or the UAV leaves the network passively due to failure, attack, and other reasons, as well as the flight obstacle that causes the UAV to temporarily leave the network; the members of the blockchain trust authority committee in each period activates the UAV flight state sensing module, sensing UAV leaving, and dynamically aggregate the corresponding UAV according to the received UAV leaving event transactions of the witness and update the cancellation identity subvector commitment, indicating the identity of the node corresponding to the revocation witness from the task network, as shown in Figure 6, UAV ID1, ID2, ID3 at due to the loss of connection state; the smart contract SC_SVCScala invokes the aggregation function module to establish or update the dual identity commitment as follows.

$$S = 1, 2, 3,$$
$$W_{123}^{\text{Revoke}} \longleftarrow \text{Aggregate}\left(C_{\text{ID}}, \text{ID}[S], \{W_i\}, i \in S.\right) \tag{3}$$

When the once departed UAV returns to the mission network, if UAV ID3 requests network communication, its identity is verified as legitimate in the first layer vector commitment, it is determined not to be a compromised node after verification in the third layer subvector commitment, and the associated smart contract then updates its second layer identity deregistration subvector commitment as follows.

$$S' = 1, 2,$$
$$W_{12}^{\text{Revoke}} \longleftarrow \text{Disaggregate}\left(W_{123}, \text{ID}[S'], \{W_j\}, j \in S'.\right) \tag{4}$$

De-aggregation with identity subvector commitment adapts to network scalability and reduces invalid communication. Revocation aggregation refers to the algorithm 2, where actively departing UAVs send departure transactions to the current authority committee; meanwhile, the authority committee members periodically sense all current trusted members of the UAV network. If no response is received for more than two periods, the unresponsive UAVs are set to leave the network state. The authority committee members in the current cycle accumulate the departure time, update the trust vector in the blockchain, and reach consensus on whether the UAV leaves the network by voting. The high four bits of the UAV trust value vector in the block structure are the trust value of the UAV, and the low four bits are the cumulative value of the time the UAV is off the network.

4.4. Untrustworthy Node Identity Subvector Commitment. To secure the entire UAV network and prevent malicious nodes from causing unbearable malicious damage to the entire network system, the triple identity subvector promises an irrecoverable revocation mechanism for malicious drone node identities. The objective is to discover and isolate the malicious nodes from the mission UAV network in the shortest possible time. The trustworthiness of the UAV nodes involved in the mission execution is guaranteed. This paper focuses on stateless local lightweight authentication based on vector commitment, node trustworthiness control refers to WatchDog algorithm to identify whether neighboring nodes are abnormal by nodes monitoring their neighboring nodes to forward packets, and the detailed process refers to [28].

The trust level saved by the UAV trust vector in the latest block is an important reference standard when selecting new

authority members in the periodic consensus. If the trust value of UAV ID_4, ID_5, ID_i is less than the threshold value set by the system, a triple identity subvector aggregation, and malicious node blacklist, an irreversible identity witness aggregation is established or updated, and the smart contract invokes the following functional module to achieve it.

$$\begin{aligned} &S = 4, 5, k, \\ &W_{45}^{\text{Blacklist}} \longleftarrow \text{Aggregate}\left(C, \text{ID}[S], \{W_i : i \in S\}\right). \end{aligned} \tag{5}$$

Triple subvector commitment: Identity witness of a node whose identity is legitimate but not trusted can be classified as a third layer of blacklisted subvector commitment. During this period, a new block is created by a bookkeeper elected by the committee and the new block is multicast with updated trust vectors and blacklisted subvector commitments to UAVs that the blockchain confirms are valid. When a UAV initiates a communication request, the UAV that receives the request first performs the first layer of vector commitment verification to determine whether the identity of the requesting node is legitimate and again verifies that its identity is trustworthy. All the verification is done locally without traversing the blockchain to query. The details are described in Algorithm 3.

4.5. Local Two-Way Authentication of UAV Node. Two-way authentication process: The identity vector commitment ensures the infeasibility of forgery attacks, man-in-the-middle attacks; timestamp mechanism ensures that reentry attack requests are directly abandoned, circumventing the formation of broadcast storms; at the same time, the random number r is generated by the initiating request node, then signed by the receiver, and sent back to the requester, confirming that it is a response to the requester's request, while the information replied by other receivers is directly rejected. The authentication protocol in the recommended scheme, whether it is a replay attack of the legitimate identity of the compromised node, or a replay attack of the external malicious node after eavesdropping, can be effectively circumvented.

Figure 7 shows an authentication process between two nodes of the task-oriented UAV network. The UAV ID_A broadcasts an authentication request, and the UAV ID_B receives the request, verifies the legitimacy of ID_A through the authentication smart contract of the local blockchain, determines the legitimacy of its identity through triple subvector commitment, detects the timestamp, and filters the replay request. After the verification is passed, ID_B sends a response to ID_A, and ID_A also verifies the legitimacy of ID_B. After passing the verification, it stops receiving the response information sent by other nodes, establishes the session key, encrypts the sent data, and sends it directly to ID_B, completing one-time transmission, where t_{A_R} is the request timestamp, t_B is the response timestamp, $r \leftarrow Z_n^*$ is the random number generated when ID_A requests, $\text{sign}(SK_A(r \| t_A)$ is the signature when UAV ID_A requests, $\text{sign}(SK_B(r \| t_B)$ is the signature when UAV ID_B responds, SK_A/PK_A, SK_B/PK_B are the public and private keys of UAV ID_A and ID_B, respectively, and W_A, W_B are the respective identity witnesses.

5. System Analysis

5.1. Authentication Correctness. Symbol explanation: The UAV network node identity vector $\overrightarrow{\text{ID}} = (\text{ID}_1, \text{ID}_2, \ldots, \text{ID}_N)$, $\overrightarrow{\text{ID}}[S] = (\text{ID}_i, i \in S)$ denotes the identity subvector represented by the ordinal number in the UAV identity set S. Using $\text{ID}[-i]$ to represent $\text{ID}[N] \setminus i$ denotes the removal of the unmanned node corresponding to position i from the identity vector. n is an integer and using $[N]$ to represent the set$\{1, 2, \ldots, N\}$. Algebraic group model means that the group elements of the adversary output cannot be created arbitrarily, but must be obtained by group computation based on the group elements. If the adversary algorithm is given group elements $X_1, X_2, \ldots, X_N \in G_1$, then each adversary algorithm outputs group elements:

$$\begin{aligned} &Z \in G_1, Z = \prod_{i=1}^{N} X_i^{Z_i}, \\ &Z_1, \ldots, Z_N \in \mathbb{Z}_p. \end{aligned} \tag{6}$$

Security assumption: Let G_1, G_2 be cyclic additive groups and G_T be cyclic multiplicative groups, both of order prime q. G_1, G_2, G_T is based on the hyperelliptic curve public key cryptosystem and satisfies the nondegenerate bilinear pairing:

$$e: G_1 \times G_2 \longrightarrow G_T. \tag{7}$$

$g_1, g_2, g_T := e(g_1, g_2)$ then are G_1, G_2, G_T generating elements, respectively. It is difficult to solve the l-wBDHE (weak bilinear Diffie–Hellman exponent problem) in the group of bilinear pairings; i.e., the probability expressed by the following equation can be neglected.

$$\begin{aligned} &\Pr\left[\left(\begin{matrix} \forall \alpha \leftarrow \mathbb{Z}_p \\ g_1^{\alpha^1}, g_1^{\alpha^2}, \ldots, g_1^{\alpha^N}, g_1^{\alpha^{N+2}}, \ldots, g_1^{\alpha^{3N}}, g_2^{\alpha^1}, g_2^{\alpha^2}, \ldots, g_2^{\alpha^N} \end{matrix}\right) : g_1^{\alpha^{N+1}}\right] \\ &= \text{negl}(\lambda), \end{aligned} \tag{8}$$

where $\alpha \leftarrow \mathbb{Z}_p$ is the secret value, no one knows after the initial generation of public parameters, the public parameters are taken from the group G_1 with $2N - 1$ values except $g_1^{\alpha(N+1)}$, and N values are taken in G_2 by calculating the values in G_T:

$$\begin{aligned} g_T^{\alpha^{(N+1)}} &= e\left(g_1^{\alpha^1}, g_2^{\alpha^1}\right) \\ &= e(g_1, g_2)^{\alpha^{(N+1)}}. \end{aligned} \tag{9}$$

Stateless verification: Establish the commitment, vector $\overrightarrow{\text{ID}} = (\text{ID}_1, \text{ID}_2, \ldots, \text{ID}_N) \in \mathbb{Z}_p^N$, and compute the commitment:

$$C_{\text{ID}} = g_1^{\sum_{i=1}^{N} ID_i \alpha^i}. \tag{10}$$

Input: Identity vector commitment, C_{ID}, related UAV ID, aggregation flag.
Output: Aggregation of the uncontacted UAVs' proof.

```
# assign committee members, monitoring all UAVs' fly status.
W^revoke = 0; Monitor_period = 5 s;
#counter: detect if UAV is online.
timeout_count = 0;
TimeoutList = 0;
# mission executing phase, crs are hardcode;
thread_monitor_leaving_Event( ) #monitoring start.
while 1 do
    #activating leaving UAV request
    Receive(ActiveleavingMsg)
    #calculating uncontacted times
    ModifyTimeoutlist( )
    if Aggregation Flag then
        W_S^Revoke ⟵ Aggregate(C, ID[S], W_i), i ∈ S
    else
        W_{S,S'}^Revoke ⟵ Disaggregate(W_123, ID[S'], W_j), j ∈ S'
    end if
end while
while aggregation flag is true do
    iftimeout_court++ > Monitor_periodthen
        blockchain.Broadcast_Send(online_hello)
        timeout_court = 0;
    end if
end while
#current turn expired,
In the header of committee:
blockchain.create(newblock)
blockchain.append(newblock)
blockchain.broadcast(newblock)
In UAV nodes:
for uav_i in length of ID⃗ do
    #when receiving new block from authority committee
    if new_block is constructed correctly then
        block_chain.append(newblock)
    else
        abort new block
    end if
end for
```

Algorithm 2: Second subvector commitment build/update.

Generate witness and member ID_i existence evidence establishment:

$$W_i = g_1^{\sum_{j\neq i} ID_j \alpha^{N+1-i+j}} = \left(\frac{C_{\mathrm{ID}}}{g_1^{\mathrm{ID}_i \alpha^i}}\right)^{\alpha^{N+1-i}}. \tag{11}$$

Member verification, based on commitment C and witness W_i verification, is

$$e\left(C_{\mathrm{ID}}, g_2^{\alpha^{N+1-i}}\right) = e\left(W_i, g_2\right) \cdot g_T^{\mathrm{ID}_i \alpha^{N+1}}. \tag{12}$$

5.2. Security Analysis. The timeliness of mission-oriented UAV networks is the biggest feature that distinguishes them from other self-organized networks. The security configuration of network nodes, such as public and private keys, and identity IDs, is generated by the mission and expires with the completion of the mission. Therefore, physical attacks such as capture and cloning are not considered, but they must have the ability to resist unauthorized access, eavesdropping, impersonation, replay, and man-in-the-middle attacks. Since the registration server that keeps the system master key does not participate in the task execution, there is no possibility of generating legitimate malicious nodes due to the master key leakage during the mission, the generation of vector commitment and witness are also completed in the task preparation stage, and the vector commitment cryptographic accumulator has conflict-free and strong unidirectionality, so the success probability of active attackers forging witnesses by constructing false member sets is negligible.

Resistance to eavesdropping attacks: Communication between UAVs in a UAV network begins with two-way authentication, and after authentication is passed, a session key is negotiated to encrypt the information for

Input: Identities VC, C_{ID}, related uav ID.
Output: Aggregation of the uncontacted UAVs' proof.
In UAV node:
$W^{Blacklist} = 0$;
watchCycle = 10 s, ObserveCounter = 0;
#watchdog in UAV observes neighbors' behaviors,
#uavs locally analysis
#send the misbehavior to the current committee.
while 1 **do**
    AnalysisObserveData ( );
    ifObserveCounter + + > watchCycle**then**
        #create untrust transaction
        SendMisBehavor (ID, behaviorType);
        ObserveCounter = 0;
    end if
end while
In Committee members:
#In current turn the committee receives the tip-offs
VoteforalluntrustedTransaction ( );
if the uav with its trust value less than 0 or current turn expired **then**
    blockchain.create (newblock)
    ckchain.append (newblock)
    ckchain.broadcast (newblock)
end if
In UAV nodes:
for uav_i in length of $\overrightarrow{ID}$. **do**
    if current_block is constructed correctly **then**
        block_chain.append (genesis_block)
    else
        abort Genesis block
    end if
end for

Algorithm 3: Third subvector commitment.

transmission. Eavesdropping attacks alone do not cause degradation of the performance of the UAV network in the mission.

Resisting man-in-the-middle attacks: Active tampering attacks that can be launched by the man-in-the-middle role through eavesdropping attacks are rejected outright because the identity and identity witness of the vector commitment cannot be forged and the identity of the man-in-the-middle node cannot be verified by the authentication smart contract because it is not registered in the stateless blockchain. Man-in-the-middle attacks do not pose a threat to the UAV network.

Resistant to replay attacks, for replay attacks after eavesdropping, the UAV network generates a large amount of invalid communication, which will seriously affect the performance of the network. There are three main methods to resist replay attacks, timestamp, execution sequence number, and random number to ensure the freshness of requests, but execution sequence number and random number methods need to save historical data and require consensus of all nodes, which is unaffordable for lightweight drones, so this paper recommends the stateless lightweight blockchain authentication method, which uses a timestamp plus a random number side for two-way authentication to identify replay attacks, reject malicious forwarding, and avoid unnecessary communication interference.

5.3. Efficacy Analysis. In this paper, we recommend a lightweight authentication scheme based on the hyperelliptic curve cryptosystem, which has a shorter key length compared to RSA and elliptic curve cryptosystem at the same security level, and its dot product operation is faster than the bilinear pair operation. It is concluded from the [29] that the relative computational cost of the bilinear [30] pair operation is about several twenty times that of the elliptic curve dot product operation; therefore, the elliptic curve dot product algorithm is more efficient and more suitable for UAV networks with limited arithmetic power. Transferring, drones run the stateless blockchain system as full nodes, and the dynamic trust authorization proof consensus mechanism ensures the security and trustworthiness of the UAV network in each round of generating new blocks. Each authentication process record is not used as a blockchain transaction to mark whether the nodes within the drone network are valid and trustworthy in the current round by recording the dynamically aggregated identity witness subvector change values into new blocks. This not only

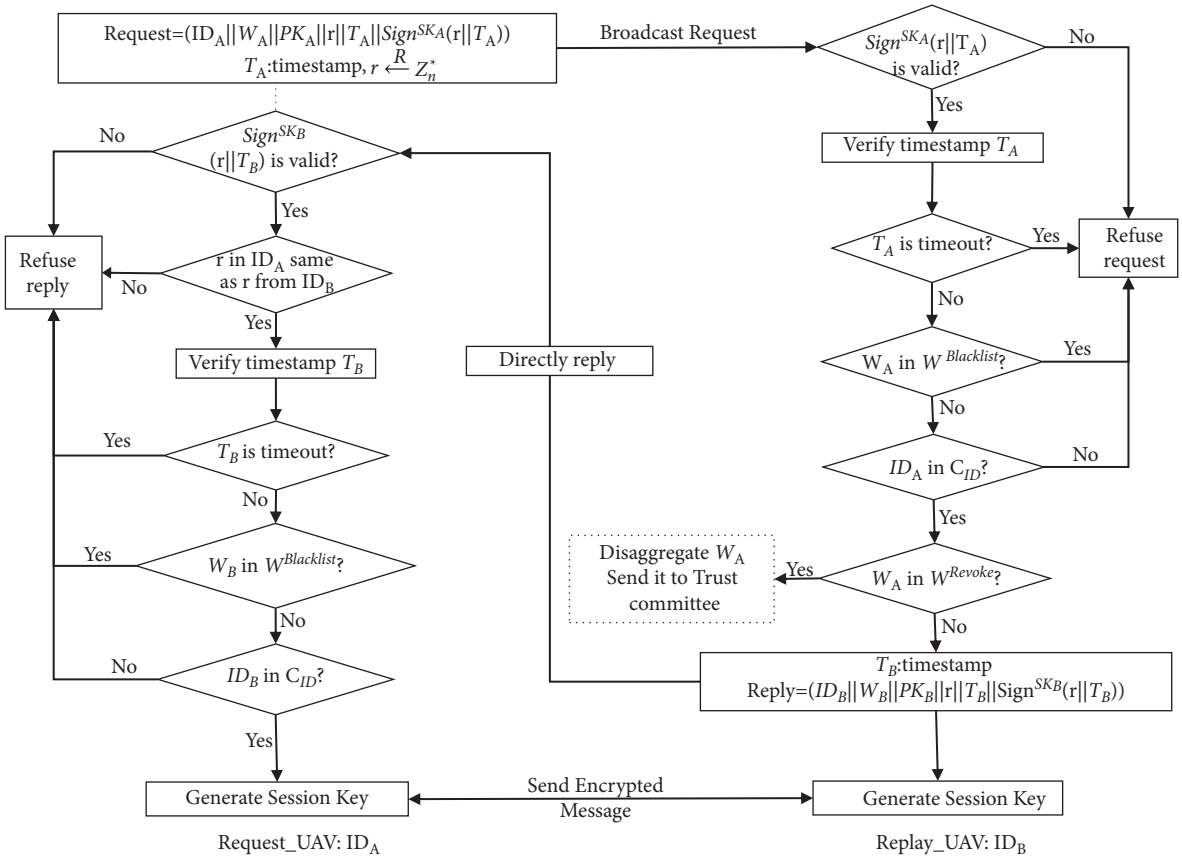


Figure 7: Communication process of the mutual authentication.

eliminates the "storage bloat" problem, but also reduces the single-step authentication time complexity from $O(n)$ to $O(\log n)$ and space complexity from $O(n)$ to $O(1)$ compared to stateful blockchain (historical state shared ledger), where no traversal of state records is required to query for authentication, but instead local authentication is performed in a proof manner. In the next section, experimental simulations and results analysis are presented in detail to effectively reduce the speed of UAV network energy consumption.

6. Experimental Simulation and Result Analysis

6.1. QualNet Network Simulation. The QualNet Simulator, developed by Scalable Networks Technologies (SNT), is software to help with network design, operation, and management. The QualNet Simulator simulates the network behavior and performance of thousands of nodes and is a comprehensive suite of tools for simulating large wireless or wired networks. The simulation experiment scenario for the proposed solution is described in Table 3. The scenario was developed by comparing the single-step authentication latency of the UAV nodes at different network sizes, the energy consumption rate of the UAV network for a fixed period of time at a specified size, the computational effort of the UAV network in the presence of different numbers of malicious nodes at a specified time (200 s), and the fixed size of the UAV network with different malicious nodes to measure the performance superiority of the stateless block authentication scheme with triple vector commitment recommended in this paper relative to the following schemes.

Scheme I [4]: relies on remote direct anonymous authentication over mobile communication link connections such as 4G: remote DAA.

Scheme II [5]: Threshold key sharing scheme.

Scheme III [18]: BlockchainPKI, a public blockchain authentication scheme for certificate tokens.

Scheme IV [25]: The stateless BlockchainVC with cryptographic accumulator.

6.2. Analysis of Experimental Results of UAV Network Simulation. Single-step authentication latency: The authentication latency is tested at the node movement speed of 10 m/s and different scales. The time required for the UAV to initiate an authentication request and obtain access or start communication after verification is passed as shown in Figure 8. In Scheme I, the UAV connects to a trusted third entity through a remote network for direct anonymous authentication, and the latency continues to increase as the number of nodes increases because all nodes share the mobile communication connection center. Scheme II increases with the size of the network and the time to collect the key share to recover the master key to ensure the security threshold value increases. In Scheme III, with blockchain certificate token authentication, the query time and

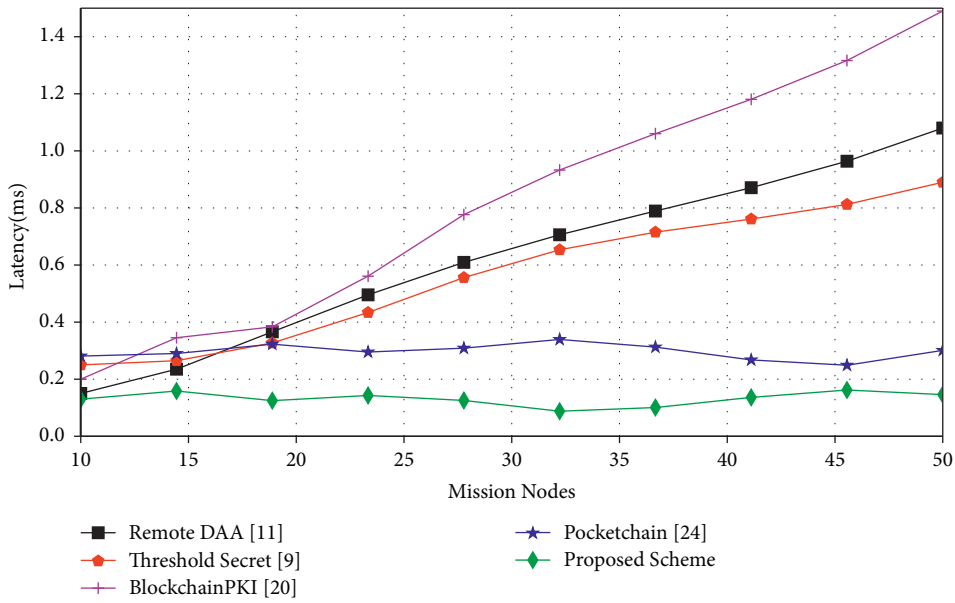


Figure 8: Single authentication latency of UAV networks at different sizes.

Table 3: Parameters related to the UAV network simulation scenario.

UAV network topology	Planar structure
Simulation area size	1000 m × 1000 m × 100 m
UAV flight speed	0, 5, 10, 15, 20, 25
Number of multicenter authorized nodes	5, 10, 15, 20
UAV node dwell time	2 s
Simulation time	800 s
Total number of UAV nodes	50
Number of lost UAV sorties triggering new blocks	2, 5, 8
Number of malicious UAV nodes	0, 5, 10, 20
New block round time (s)	10, 20

consensus time grow rapidly with the number of outgoing blockchain certificates and the increase in the size of participating network nodes. Constructing a stateless blockchain with the cryptographic accumulator approach in Scheme IV, the time for authentication is theoretically constant in magnitude, but fluctuates in time due to recalculation of accumulation values and network member witnesses caused by UAVs entering and leaving the network. The recommended method does not update computation by triple vector commitment and only changes some of the member witness aggregation to other subvectors into a promise, and the authentication delay fluctuation is small.

The rate of energy consumption of the UAV network: The consumption of the mission UAV network energy is directly related to the UAV range, and reducing the consumption rate of energy usage is the key to mission completion. Figure 9 shows the simulation test of five scenarios; in the time of 800 s, 50 UAV network, the presence of 20 malicious nodes, and the implementation of replay attack case, observe the rate of energy decline; in Scheme III due to the consensus algorithm of proof of workload, energy consumption is the fastest, about 400 s of time simulation energy is consumed; Scheme I requires remote communication, shared channel resource competition, and the interference from replay attacks; the energy consumption also decreases quickly and eventually ends around 500 s; and because the UAV moves in a random wandering manner, resulting in frequent access to the network by the UAV, leading to an increase in the computation of the update of Scheme IV, the energy decreases significantly at a later stage. Recommended scheme. The recommended scheme because they are all local authentication, no consensus, and better resistance to replay attacks, knowledge in maintaining the network trustworthy is the DPOTA consensus protocol cycle, processing can be aggregated subvector commitment operations, energy consumption is small, energy consumption is also the slowest, increasing the overall working time of the UAV network.

Computational cost under different numbers of malicious nodes: The test conditions are set up with a drone network size of 50 drones, running for 100 seconds, with different numbers of malicious nodes in the network, initiating the same communication task, and comparing the computational cost required for the five authentication schemes. As shown in Figure 10, Scheme III has insignificant changes because the computational overhead is mainly

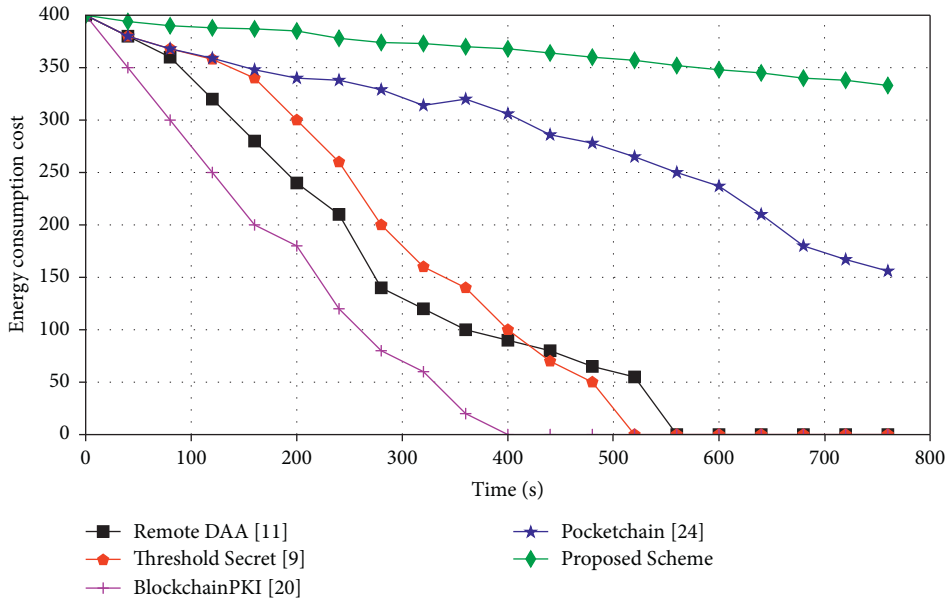


Figure 9: UAV network energy consumption rate.

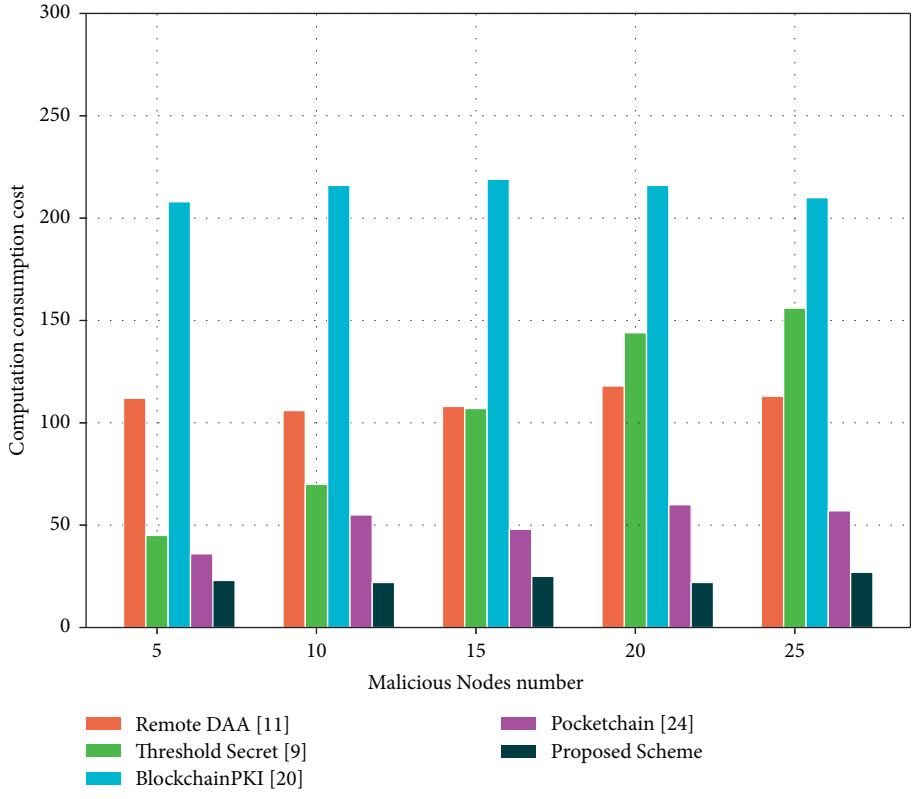


Figure 10: Computational cost with the different malicious nodes.

derived from the consensus overhead caused by the scale of the nodes due to the qualities of the traditional blockchain itself to prevent double-splash attacks; Scheme I, which relies on a remote third-party trusted entity to provide authentication, can resist replay attacks, and the computational overhead is basically unchanged; Scheme II has a rapid increase in computational overhead when the number of malicious nodes increases, as there is no effective defense given by the certificate center or blockchain platform. The computational overhead of Scheme IV also increases gradually because of the increase in malicious nodes, which increases the frequency of recalculating the cumulative value and updating the identity witness of its system.

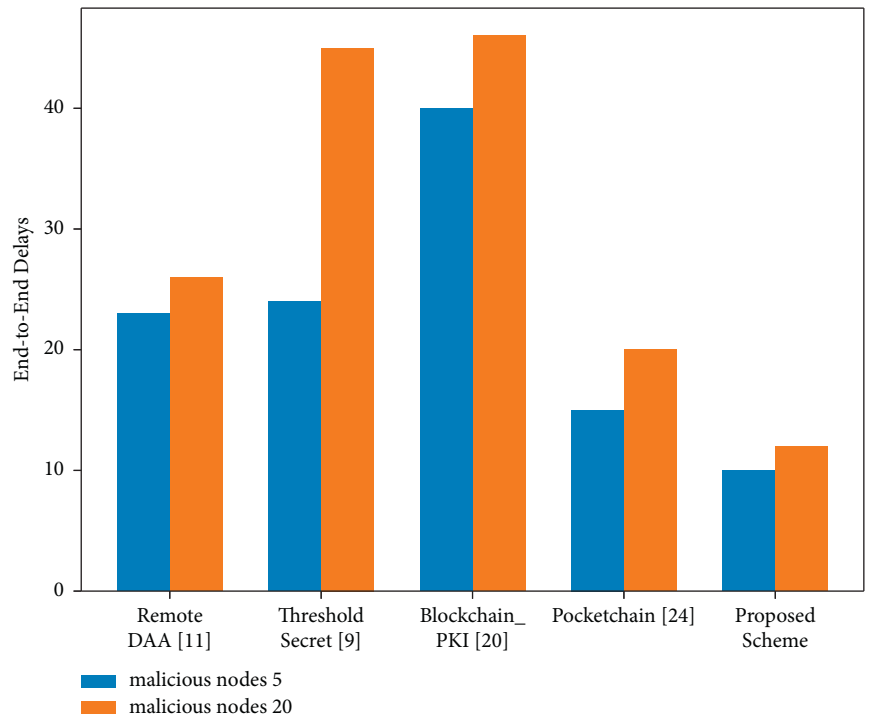


Figure 11: End-to-end transmission delay in the presence of malicious nodes.

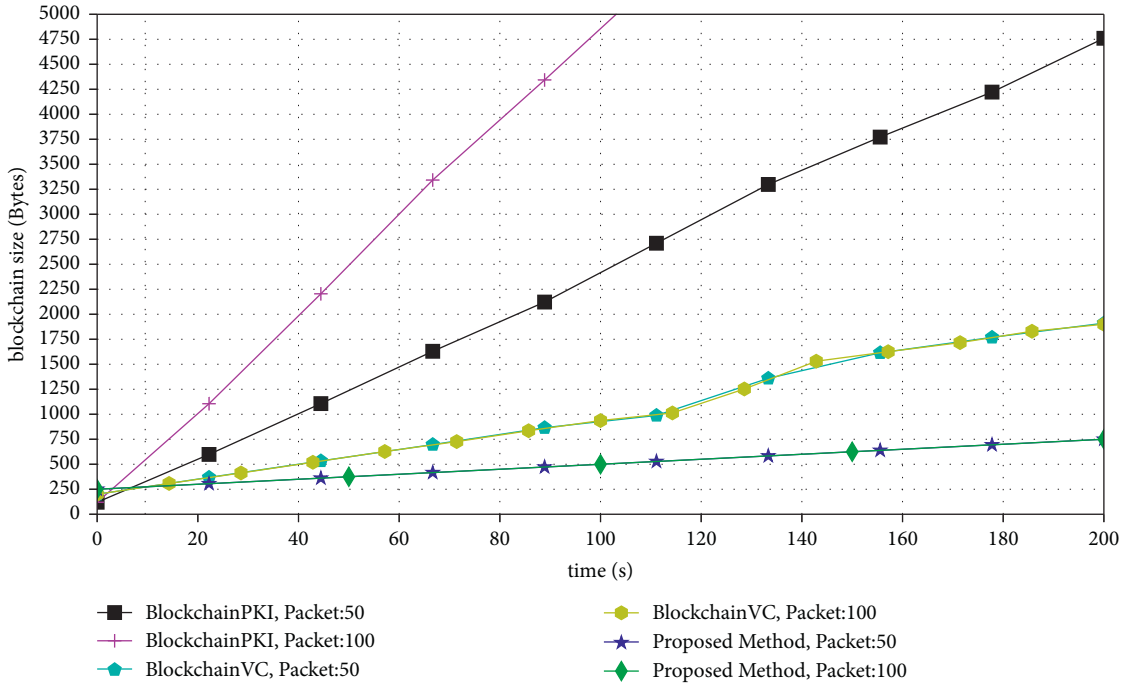


Figure 12: Comparison of storage growth of different blockchains in UAV networks.

End-to-end transmission latency under different numbers of malicious nodes: The test conditions are set with a drone network size of 50 drones and the presence of 5 malicious nodes and 20 malicious nodes in the network. The end-to-end communication latency of the five authentication schemes is compared, as shown in Figure 11. Scheme III, the interference of replay attacks by malicious nodes on end-to-end transmission, is negligible due to the traditional blockchain with the feature of preventing replay attacks, and the inefficient consensus leads to its high time consumption. Scheme I, which relies on remote third-party trusted entities to provide authentication, can resist replay attacks, and end-to-end latency makes no difference in these two cases. In Scheme II, when the number of malicious nodes increases, its end-to-end

transmission latency is severely affected due to the absence of effective defense given by certificate centers or blockchain platforms; Scheme IV, because the increase of malicious nodes leads to the change of effective nodes in the network, which increases the computation of commitment and witness updates, thus affecting the end-to-end transmission latency; Recommended scenario, due to local two-way authentication and effective defense against malicious nodes, the end-to-end changes in transmission latency are minimal.

Consensus and storage: Blockchains are shared databases that keep growing along with consensus. Experiments are conducted to compare the storage requirements of drone networks under different blockchains. To satisfy comparability, the following experimental scenario is set up, where malicious nodes are not considered, the UAV network is well connected, the network size is 100 nodes, the running time is 200 seconds, the routing protocol is DSR, all nodes send data randomly every 5 seconds, and the size of data packets is fixed.

(1) Traditional blockchain based on a distributed PKI with a delegated proof of stake consensus algorithm (DPOS). Each time a packet is sent as a transaction, consensus is accomplished by a fixed number of 21 delegated nodes, with a provision to initiate consensus every 20 seconds.

(2) Stateless blockchain based on accumulator: same as above.

(3) Stateless authentication chain recommended in this paper: set the consensus cycle to 20 s, and the local trustworthiness assessment generated by monitoring the forwarding behavior of neighboring nodes on routing information and data packets as a data consensus transaction, again reaching consensus among the 21 authorized nodes selected dynamically in the cycle and completing consensus on the decision.

The experimental results are shown in Figure 12.

As with traditional blockchains, each of transaction data needs to be on the chain, and the new block after consensus is reached contains the transaction data within 20 s. As new blocks are created, the size of the blockchain keeps increasing, and the larger the transaction data package, the faster the blockchain grows.

Stateless blockchain based on cryptographic accumulator or vector commitment is to create new blocks with authentication results as transactions, and the new blocks reach consensus at delegated authorized nodes to finally confirm the authentication success. Its transactions are smaller than the authenticated data, but still have transaction blocks.

The recommended stateless authentication chain is with triple identity vector commitment, its consensus process contains data consensus and decision consensus, the local trust assessment of all nodes to their neighboring nodes in each cycle is the object of its number consensus, its ultimate purpose is to obtain decision results through statistical analysis of the results of data consensus, its decision results in fixed size, including updated triple vector commitment and new authorized node group, the size is not more than 50 bytes. The historical state data used for data consensus do not need to be saved.

At the same time, it is clear that the first two types of authentication are confirmed after the block consensus, then the blockchain is updated, and their authentication efficiency is equivalent to the consensus efficiency. The recommended solution, on the other hand, whose consensus aims to maintain the trustworthiness of the drone network, is authenticated locally by the nodes on that blockchain's trusted platform, which is fast and not limited by the size of the network.

7. Conclusions

In this paper, a scheme for lightweight mutual authentication of UAV network nodes is proposed. The recommended scheme is based on vector commitment to establish a stateless blockchain with a consensus mechanism of dynamic multicentric trust authorization proof to maintain the trustworthiness and effectiveness of participating nodes in the UAV network during mission execution in the scenario of dynamic changes in the size and agency of the mission network due to environmental factors and cyber attacks. According to the timeliness requirements of the mission network, a triple aggregatable subvector commitment mutual authentication protocol is designed to effectively resist counterfeit attacks, man-in-the-middle attacks, and replay attacks. Simulation experiments demonstrate that this scheme has better performance in terms of energy consumption, computational cost, single authentication latency, and end-to-end delay compared to current authentication methods that can run in mission-based UAV networks.

Data Availability

Due to the privacy of the data and sensitive information, it is not convenient to provide.

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 (2019YFB2102002), in part by the National Natural Science Foundation of China (62176122 and 62001217), and in part by A3 Foresight Program of NSFC (62061146002).

References

[1] İ. Bekmezci, OK Sahingoz, and S Temel, "Flying ad-hoc networks (fanets): Flying Ad-Hoc Networks (FANETs): A survey survey," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, 2013.

[2] V. A. K. Singh K, *Threat Modeling for Multi-Uavs Adhoc Networks*, Tencon, Japan, 2017.

[3] K. Kurosawa, S. Obana, and W. Ogata, "Threshold secret sharing schemes," in *Proceedings of the Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference*, vol. 963, pp. 410–423, Springer, Santa Barbara, CA, USA, 1995.

[4] L. Chen, S. Qian, M. Lim, and S. Wang, "An enhanced direct anonymous attestation scheme with mutual authentication for network-connected uav communication systems," *China Communications*, vol. 15, no. 5, pp. 61–76, 2018.

[5] A. Alomari, "Fully distributed certificate authority based on polynomial over elliptic curve for MANET," *International Journal of Networked and Distributed Computing*, vol. 2, no. 2, pp. 70–77, 2014.

[6] Y Seung and K Robin, "MOCA: mobile certificate authority for wireless Ad Hoc networks," 2004.

[7] A. Khalili, J. Katz, and W. A. Arbaugh, "Toward secure key distribution in truly ad-hoc networks," in *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT 2003)*, pp. 342–346, IEEE Computer Society, Orlando, FL, USA, January 2003.

[8] H. Deng, A. Mukherjee, and D. P. Agrawal, "Threshold and identity-based key management and authentication for wireless ad hoc networks," in *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, vol. 1, pp. 107–111, IEEE Computer Society, Las Vegas, NV, USA, April 2004.

[9] J. Chen, J. Ling, J. Ning, and J. Ding, "Identity-based signature schemes for multivariate public key cryptosystems," *The Computer Journal*, vol. 62, no. 8, pp. 1132–1147, 2019.

[10] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proceedings of the Advances in Cryptology - ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security*, vol. 2894, pp. 452–473, Springer, Taipei, Taiwan, December 2003.

[11] J. Zheng, S. Xu, F. Zhao, D. Wang, and Y. Li, "A novel detective and self-organized certificateless key management scheme in mobile ad hoc networks," in *Proceedings of the IEEE International Conference on Granular Computing, GrC 2013*, pp. 443–448, IEEE Computer Society, Beijing, China, December 2013.

[12] Y. Zhao, Y. Hou, Y. Chen, S. Kumar, and F. Deng, "An efficient certificateless public key encryption with equality test toward internet of vehicles," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 5, 2022.

[13] E. Frimpong, R. Rabbaninejad, and A. Michalas, "Arrows in a quiver: a secure certificateless group key distribution protocol for drones," *IACR Cryptol. ePrint Arch.*, Virtual Event, November 2021.

[14] D. Mishra and S. Mukhopadhyay, "A certificateless authenticated key agreement protocol for digital rights management system," in *Proceedings of the Quality, Reliability, Security and Robustness in Heterogeneous Networks - 9th International Conference*, vol. 115, pp. 568–577, Springer, Greader Noida India, January 2013.

[15] Y. Tan, J. Liu, and N. Kato, "Blockchain-based key management for heterogeneous flying ad hoc network," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7629–7638, 2021.

[16] M. Toorani and C. Gehrmann, "A decentralized dynamic PKI based on blockchain," 2020, https://arxiv.org/abs/2012.15351.

[17] M. Y. Kubilay, M. S. Kiraz, and H. A. Mantar, "Certledger: a new PKI model with certificate transparency based on blockchain," *Computers & Security*, vol. 85, p. 1071, 2018.

[18] Q. T. Thai, J. Yim, and S. Kim, "A scalable semi-permissionless blockchain framework," in *Proceedings of the 2019 International Conference on Information and Communication Technology Convergence*, pp. 990–995, IEEE, Jeju Island, Republic of Korea, October 2019.

[19] J. C. Benaloh and M. de Mare, "One-way accumulators: a decentralized alternative to digital sinatures (extended abstract)," in *Proceedings of the Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques*, vol. 765, pp. 274–285, Springer, Lofthus, Norway, May 1993.

[20] D. Boneh, B. Bünz, and B. Fisch, "Batching techniques for accumulators with applications to iops and stateless blockchains," in *Proceedings of the Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference*, vol. 11692, pp. 561–586, Springer, Santa Barbara, CA, USA, August 2019.

[21] F. Baldimtsi, R. Canetti, and S. Yakoubov, "Universally composable accumulators," in *Proceedings of the Topics in Cryptology - CT-RSA 2020 - The Cryptographers' Track at the RSA Conference 2020*, vol. 12006, pp. 638–666, Springer, San Francisco, CA, USA, February 2020.

[22] D. Catalano and D. Fiore, "Vector commitments and their applications," in *Proceedings of the Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography*, vol. 7778, pp. 55–72, Springer, Nara, Japan, February 2013.

[23] M. Campanelli, D. Fiore, N. Greco, D. Kolonelos, and L. Nizzardo, "Vector commitment techniques and applications to verifiable decentralized storage," 2020, https://eprint.iacr.org/2020/149.

[24] A. Tomescu, I. Abraham, V. Buterin, J. Drake, D. Feist, and D. Khovratovich, "Aggregatable subvector commitments for stateless cryptocurrencies," in *Proceedings of the Security and Cryptography for Networks - 12th International Conference, SCN 2020*, vol. 12238, pp. 45–64, Springer, Amalfi, Italy, September 2020.

[25] R. W. F. Lai and G. Malavolta, "Subvector commitments with application to succinct arguments," in *Proceedings of the Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference*, A. Boldyreva and D. Micciancio, Eds., vol. 11692, pp. 530–560, Springer, Santa Barbara, CA, USA, August 2019.

[26] V. Keerthika and R. Suganthe, "Watchdog: reduce time delay for spreading selfish information in manet," in *Proceedings of the 2013 International Conference on Information Communication and Embedded Systems*, pp. 1104–1107, Chennai, India, 2013.

[27] U. Ali, M. Y. I. B. Idris, M. N. B. Ayub et al., "Rfid authentication scheme based on hyperelliptic curve signcryption," *IEEE Access*, vol. 9, Article ID 49942, 2021.

[28] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar, and K.-K. R. Choo, "Homechain: a blockchain-based secure mutual authentication system for smart homes," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 818–829, 2020.

[29] Y. Liao, Y. Liu, Y. Liang, Y. Wu, and X. Nie, "Revisit of certificateless signature scheme used to remote authentication schemes for wireless body area networks," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2160–2168, 2020.

[30] F. Guo, Y. Mu, W. Susilo, H. Hsing, D. S. Wong, and V. Varadharajan, "Optimized identity-based encryption from bilinear pairing for lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 2, pp. 211–220, 2017.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 7302767, 9 pages
https://doi.org/10.1155/2022/7302767

Research Article

Improved Efficient Privacy-Preserving Certificateless Provable Data Possession Scheme for Cloud Storage

Xiuguang Li,[1,2] **Ruifeng Li**,[2] **Xu An Wang**,[2] **Ke Niu**,[2] **Hui Li**,[1] **and Xiaoyuan Yang**[2]

[1]*State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, China*
[2]*Chinese People's Armed Police Force Engineering University, Xi'an, China*

Correspondence should be addressed to Hui Li; lihui@mail.xidian.edu.cn

Received 2 June 2022; Accepted 15 July 2022; Published 30 August 2022

Academic Editor: Yinbin Miao

Cloud storage technology is evolving at a high speed; effectively auditing the cloud data's integrity has become a focal point. Recently, Ming and Shi proposed a certificateless integrity auditing scheme with a privacy protection function. The scheme used the certificateless cryptosystem to solve the certificate management problem of the auditing schemes based on public key infrastructure and the key escrow problem of the identity-based auditing schemes. Although their scheme is novel and efficient, we found that their scheme was not secure and could not achieve integrity auditing of cloud data. The malicious cloud server can generate the proof through the blocks and tags sent by the user. On the basis of the original scheme, we propose an improved auditing scheme; our new scheme is more secure and effective. In addition, for the problem of idle tags in the existing cloud data integrity auditing scheme, we propose the idea of intermediate tags and we applied the idea to the improved scheme to improve audit efficiency.

1. Introduction

Users are increasingly inclined to store data in the cloud to obtain more convenient data management services. Cloud service providers (CSP) centrally hold massive amounts of users' data. For an attacker, a successful attack on the cloud server will gain a great deal. Therefore, it is easy for CSPs to become the targets of centralized attacks. The dishonest CSPs may also deliberately delete users' data to reduce their own storage burden or deliberately conceal security incidents that damage data integrity to maintain their own reputation. Therefore, the cloud data integrity audit schemes are proposed to effectively solve problems [1].

Motivation: We note that the existing audit schemes require users to calculate data tags corresponding to all data blocks when preprocessing data blocks and upload them to the CSP for storage. However, in the auditing process, generally few tags are used to generate the proof. Once the proof is verified, it can ensure that each data block specified by the auditor is complete and guarantee all original data's integrity with a high confidence probability. For 1,000,000 data blocks with a size of 4 kB, assuming that the server deletes or is tampered with 1% of the data blocks, the auditor only needs to audit 460 data blocks, which can be higher than 99% confidence probability [2] to judge the integrity of all the data. Therefore, most of the data tags are idle during the audit process. Suppose it is an application scenario where data blocks are frequently updated [3]; a large number of tags are calculated and stored in the cloud, but they will be updated as the data blocks are updated before they are used, resulting in larger computing and storage resources waste. To solve the idle tags problem, we propose the idea of intermediate tags. Before uploading the data, when processing the data, users only generate the key intermediate tags and then upload them to CSP. When the third-party auditor (TPA) challenges the cloud data, CSP generates complete certification tags for the challenged data block. Then they enter the normal audit process.

Recently, Ming and Shi [4] proposed a certificateless auditing scheme called CLPDP that supports privacy protection. In their scheme, CSP can use tags to easily forge the proof. Even if all outsourced data are deleted by CSP, it can

still give the correct proof to pass the audit. So we point out the security problem in their scheme. In addition, we find that the original scheme is one that can apply the idea of intermediate tags. Therefore, we also improved the original scheme.

Our contributions are as follows:

(1) We analyze Ming and Shi's scheme and find the security problem. CSP can forge the proof to pass the audit. Then we described the attack method in detail.

(2) We propose the idea of intermediate tags, which can reduce the computing overhead of users in audit schemes. After improving the safety of the original scheme, we use the idea of intermediate tags to promote the original scheme.

(3) We performed security analysis on the improved scheme, and we proved that the improved scheme is secure. The efficiency of the improved scheme is also analyzed and compared with that of the original scheme. The improved scheme is more efficient, which proves the applicability of intermediate tags.

2. Related Works

Early data integrity audit schemes required users to download all their stored data and verify the downloaded data locally. However, most users store a large amount of data, so it requires high communication, storage, and computing costs for users to download all data for verification, and users generally cannot meet such requirements. Ateniese et al. [2] formally defined the Provable Data Possession (PDP) scheme. When verifying the integrity, users divide data files into blocks, and only partial data blocks are downloaded. Finally, the integrity of all data can be verified with a very high confidence rate. This method enables users to complete the audit task without downloading complete files, reducing the huge communication cost in the process of a data integrity audit. In 2013, Wang et al. [5] introduced TPA into the data integrity audit system. Users can further reduce their own expenses by outsourcing audit tasks to TPA. At present, scholars add various functions to the basic data integrity audit scheme [2] to meet the requirements of different application scenarios.

Users will inevitably need to change their data after uploading data files. Therefore, the cloud data's content should be allowed to change dynamically. Considering the urgent need for data integrity audit schemes in the dynamic update, scholars put forward audit schemes with dynamic update functions. Dynamic data update has gradually become the basic function in cloud data integrity audit schemes, which is indispensable in the application of real scenarios. Existing data structures applied to dynamic data updates mainly include Index Switcher, Index Hash Table, Merkle Hash Tree, Skip List, Dynamic Hash Table, Red-Black Tree, etc. In the construction of a data integrity verification scheme supporting dynamic data updates, the difficulty lies in solving the problem of extra computation costs caused by index change.

Jin et al. [6] constructed the mapping from the data block index to the tag index and designed the Index Switcher data structure to avoid the extra computational overhead caused by tag recalculation. In addition, the dispute arbitration function is added to the proposed audit scheme to ensure that users or the cloud will not commit improper acts during the audit process. Tian et al. [7] proposed the audit scheme supporting dynamic updates, privacy protection, and batch audit. The dynamic hash table data structure is designed to realize fast audit and efficient data updates by recording the attributes of files and data blocks at the audit ends. Shen et al. [8] proposed the whole/sampling audit method to solve the problem of distrust between users and the cloud and designed a double-linked information table to achieve efficient data update. Their scheme also supported the batch audit function. Guo et al. [9] constructed a multileaf authentication method based on the Merkle tree, which can simultaneously authenticate multiple leaf nodes and corresponding indexes and realize batch data updates. The scheme supports log auditing. By checking the log files generated by auditors, users can verify whether the auditors perform their audit work honestly. The public audit protocol designed by Hou et al. [10] supports blockless verification and batch verification. The scheme uses the chameleon authentication tree to realize the efficient and dynamic operation of outsourced data and reduces computing costs and improves the audit efficiency. Mishra et al. [11] used a binomial binary tree and indexed hash table data structure to construct an audit scheme supporting batch audit and efficient dynamic update based on BLS signature.

The reliability of data is the basis of its value and benefit. After the reliability of data is solved, other problems of data such as consistency, practicality, and availability are meaningful. Multicopy storage is the most straightforward and simple way to improve reliability. CSP provides storage services at low prices. Users can use the massive storage space it provides. More and more users choose multicopy storage to obtain more availability of data. The audit schemes supporting dynamic manipulation of multiple replicas while ensuring data integrity remain to be explored and further investigated. Curtmola et al. [12] constructed the first multicopy audit scheme, in which each copy can generate a corresponding integrity proof against challenges, and storing multiple copies is more efficient than storing each copy individually. Liu et al. [13] constructed the multiple-copy audit scheme supporting data dynamic updating. The Merkle hash tree node used in their scheme contains the node level parameters, which are allocated to each data block. It is more efficient when verifying multiple replica updates. The audit scheme of Guo et al. [14] reduces the storage burden of CSP by sharing an authenticated identity tree among multiple copies. The scheme supports multicopy and batch auditing, which also reduces the computational cost. Yaling and Li [15] proposed a flexible multicopy PDP scheme based on the characteristics of a multibranch tree. Their scheme ensures the integrity and reliability of multiple copies and implements the verification of any copy and supports dynamic update operation and privacy protection.

In recent years, in order to optimize audit performance and improve update efficiency, batch audit and batch update have become indispensable functions of cloud data integrity audit schemes. Qi et al. [16] applied the rank-based Merkle hash balanced tree to integrity verification and improved the dynamic update's efficiency. Deng et al. [17] implemented batch auditing using BLS signature and rank-based Merkle hash tree.

Later, scholars introduce TPA to perform a public audit on behalf of users to reduce the computation cost. However, TPAs are often not fully trusted [18], which can lead to the disclosure of users' privacy [19]. Li et al. [20] solved the key management problem based on fuzzy identity. The scheme took the user's biometrics as the identity and designed a corresponding audit protocol to protect the data content. Wang et al. [21] scheme uses a ring signature to calculate the metadata required for verification. The authenticator and random mask technology are used to protect data privacy; the scheme can also realize batch audits. The audit scheme of Wang et al. [22] is based on an algebraic signature and integrates forward error correction codes to enhance data possession assurance and recover data when a small number of blocks are deleted, thus significantly reducing communication complexity.

With the development of blockchain technology, many scholars apply blockchain technology to cloud data integrity audits [23]. The certificateless audit scheme proposed by Zhang et al. [24] can resist malicious TPA; the scheme uses Bitcoin as the source of pseudorandom numbers to help generate challenging information. Li et al. [25] proposed a lightweight audit scheme with blockchain technology for integrity audit. In their scheme, the user and CSP are set as two mutually untrustworthy entities, and the TPA is removed. After the user stores the lightweight verification tags into the blockchain, the Merkle hash tree is constructed through the tags to generate the proof, so as to save computational power. Yang et al. [26] provided the mutual blockchain for outsourced cloud data and proposed an incentive mechanism based on credit, so that CSPs can supervise each other, which prevents collusion and realizes public audit efficiently. Yang et al. [27] proposed a certificateless multicopy and multicloud data public audit scheme based on blockchain technology. Their scheme leverages the unpredictability of blocks in the blockchain to build fair challenge information, preventing malicious auditors from colluding with CSP to deceive users. Wang et al. [28] used blockchain to replace TPAs and designed a blockchain-based fair payment smart contract for a cloud data audit. In their scheme, users and CSP will run blockchain-based smart contracts to ensure that the cloud periodically submits data to the cloud with proof of possession. Only after verification can the CSP be paid. Wei et al. [29] built a blockchain integrity protection mechanism. The scheme deploys the distributed virtual machine agent model on the cloud allowing multitenant collaboration and achieving reliable storage, monitoring, and verification tasks. Reference [30] proposed a protection model based on a private chain, which synchronously uploads modification records of files and hash values of files to blockchain for storage and judges whether the data is complete by comparing hash values.

Quantum computers use qubits to represent many possible states of 1 and 0 at the same time and have more processing power than standard computers. Most cloud storage data auditing schemes are based on a traditional cryptosystem. However, with the introduction of algorithms such as quantum large number decomposition, the traditional cryptosystem loses its security. Lattice-based cryptography is generally considered to be effective against the quantum attack. Xu et al. [31] designed the first lattice-based cloud data audit scheme based on the small integer solution problem. The audit scheme designed by Liu and Cao [32] supports public verification but does not provide strict security certification. Zhang et al. [33] designed an ID-based public audit protocol based on lattice by using ID-based signature technology and further provided a solution to solve the key exposure problem [34], which protected user data's privacy. In addition, TPA cannot obtain information about users' data during audit verification. Sasikala and Shoba Bindu [35] designed a lattice-based certificateless public auditing protocol for the first time, but it was pointed out by [36] that the scheme had security problems.

Organization: We organize our paper as follows. In Section 3, we reviewed the certificateless privacy protection secure cloud storage scheme of Ming and Shi. Section 4 describes the attack against the original scheme. In Section 5, we propose the concept of intermediate tags and give an improved audit protocol. In Section 6, the security and performance of the improved scheme are analyzed to prove that it is safer and more efficient. Finally, in Section 7, we summarize our work.

3. Review of Ming and Shi's Scheme

The system model of Ming and Shi is shown in Figure 1, including a key generation center (KGC), a data owner (DO), CSP, a data user (DU), and TPA. Figure 1 shows their system model. To facilitate understanding, we define and explain the various symbols and variables that appear in our paper in Table 1.

Specifically, the following is the operation process of the original scheme:

(1) Setup: KGC first selects the cyclic group G on the elliptic curve E, defines the large prime number q with the order of G, and selects the generator $P \in G$. Then it selects the secure hash function $H_{1,2,3,4}: \{0,1\}^* \longrightarrow Z_q^*$, selects a random $\lambda \in Z_q^*$ as the system master key, and calculates $P_{\text{pub}} = \lambda \cdot P \in G$. Finally, KGC keeps the master key in secret and exposes the parameters.

(2) PartialKeyGen: DO sends the real identity information $\text{ID} \in Z_q^*$ to KGC. After KGC receives $\text{ID} \in Z_q^*$, it selects a random number $u \in Z_q^*$ and calculates $PID_1 = u \cdot P$, $PID_2 = I\ D \oplus H_1(u \cdot P_{\text{pub}} \| PID_1)$.

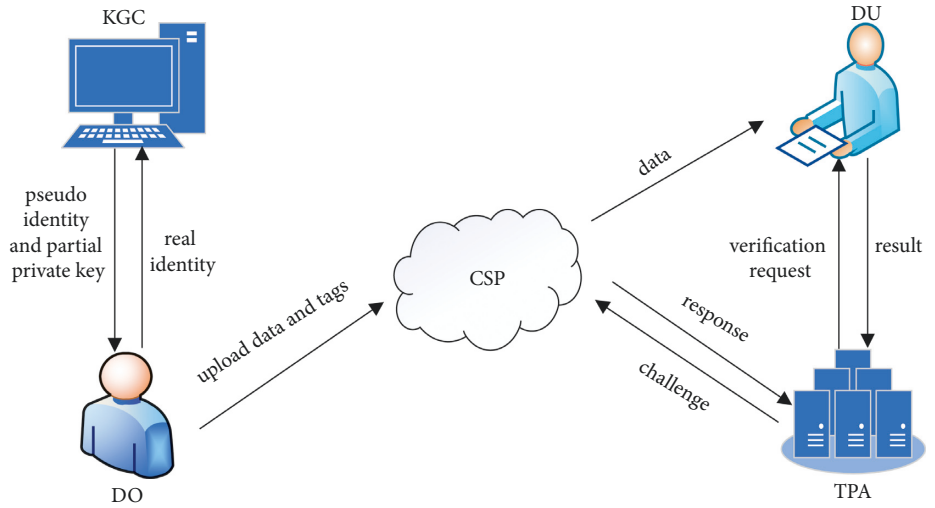


Figure 1: The system model.

Table 1: Notations.

Notations	Descriptions
E	The elliptic curve
F_p	The finite field
G	The cyclic group
P	Generator of G
$H_{1,2,3,4}$	Secure hash function $H(\cdot)$: $\{0,1\}^* \longrightarrow Z_q^*$
P_{pub}	Public key of the system
$\text{ID} \in Z_q^*$	DO's real identity
PID	DO's virtual identity
λ, u, d, r_i, v_i	Random numbers
(x, y)	DO's secret key
(D, X)	DO's public key
id_i	The identifier of the data block m_i
m_i	The data block
ω_j, ϕ_j, τ	Intermediate parameters
R_i, s_i	The tags of m_i
Q	The collection of challenged indexes
$\{\alpha, \beta\}$	The proof
H	The computational cost of one hash
A_Z	The computational cost of one addition on Z_q^*
M_Z	The computational cost of one multiplication on Z_q^*
A_G	The computational cost of one point addition on G
M_G	The computational cost of one point multiplication on G

Then KGC sends DO's virtual identity $PI\ D = \{PID_1, PID_2\}$ to DO and randomly selects $d \in Z_q^*$.

alculates $D = d \cdot P$, $\tau = H_2(PI\ D \| D)$, $y = d + \lambda \cdot \tau$. Finally, KGC sends DO's partial keys $\{D, y\}$ to DO.

(3) SecretValueGen: DO randomly chooses $x \in Z_q^*$ and obtains complete private key $\{x, y\}$.

(4) PublicKeyGen: DO calculates $X = x \cdot P$ and obtains the complete public key $\{D, X\}$.

(5) TagGen: the data file M is divided into n blocks by DO as $M = \{m_1, m_2, \ldots, m_n\}$, where $m_{i(1 \le i \le n)} \in Z_q^*$. DO selects a random number $r_i \in Z_q^*$ and calculates $R_i = r_i P$, $w_i = H_3(X \| R_i \| id_i)$, $\phi_i = H_4(D \| R_i \| id_i)$, and $s_i = r_i \cdot m_i + w_i \cdot x + \phi_i \cdot y$ for $i \in \{1, 2, \ldots, n\}$, where id_i is the identifier of m_i. Thus the tags $\sigma = \{R_1, R_2, \ldots, R_n, s_1, s_2, \ldots, s_n\}$ are generated by DO; they are sent with the data blocks to CSP. DO deletes the local data and tags.

(6) Challenge: after receiving DU's audit request, TPA generates the challenge message. It first selects a random subset Q in $\{1, 2, \ldots, n\}$. The subset Q includes c elements. For $j \in Q$, TPA randomly selects $v_j \in \mathbb{Z}_q^*$; then it sends $\text{chal} = \{j, v_j\}_{j \in Q}$ as the challenge message to CSP.

(7) ProofGen: CSP calculates $\alpha = \sum_{j \in Q} v_j \cdot s_j \cdot P$ and $\beta = \sum_{j \in Q} v_j \cdot m_j \cdot R_j$ after it receives $\text{chal} = \{j, v_j\}_{j \in Q}$; then it sends $\{\alpha, \beta\}$ as the proof to TPA.

(8) Verify: after receiving $\{\alpha, \beta\}$, TPA calculates $\tau = H_2(PI\ D \| D)$. Then it calculates $\omega_j = H_3(X \| R_j \| id_j)$ and $\phi_j = H_4(D \| R_j \| id_j)$ for $j \in Q$ and verifies

$$\alpha \stackrel{?}{=} \beta + \left(\sum_{j \in Q} \omega_j \cdot v_j\right) \cdot X + \left(\sum_{j \in Q} \phi_j \cdot v_j\right) \cdot \left(D + \tau \cdot P_{\text{pub}}\right). \tag{1}$$

If equation (1) holds, DO's data is complete.

The proof of the correctness of equation (1) is as follows:

$$\begin{aligned}\alpha &= \sum_{j\in Q} v_j s_j P = \sum_{j\in Q} v_j\left(r_j m_j + \omega_j x + \phi_j y\right)P \\ &= \sum_{j\in Q} v_j r_j m_j P + \sum_{j\in Q} v_j \omega_j x P + \sum_{j\in Q} v_j \phi_j y P \\ &= \sum_{j\in Q} v_j m_j R_j + \left(\sum_{j\in Q} v_j \omega_j\right)X + \left(\sum_{j\in Q} v_j \phi_j\right)\left(D + \tau P_{\text{pub}}\right) \\ &= \beta + \left(\sum_{j\in Q} \omega_j v_j\right)X + \left(\sum_{j\in Q} \phi_j v_j\right)\left(D + \tau P_{\text{pub}}\right).\end{aligned} \tag{2}$$

4. Our Attack

In the scheme of Ming and Shi, we find that the CSP can calculate the value of the aggregated data blocks needed at the Proofgen stage. In this way, even if the CSP deletes DO's cloud data, the correct data possession proof can be generated by it at the Proofgen stage and passed the audit. In this section, we show two types of attacks; the process by which CSP forges the "correct" blocks is also introduced.

4.1. The First Type of Attack. The first attack is caused by a design error in the verification equation; the detailed description is as follows:

Assume that the entities in the scenario run the audit scheme following the process described above; when the scheme progresses to the ProofGen stage, CSP needs to generate the proof $\{\alpha, \beta\}$. We note that, in equation (1), CSP can obtain all values except α and β, so CSP just needs to randomly select $\alpha \in G$; it can obtain β by calculating equation (1). Similarly, CSP can also calculate the value of α by calculating equation (1) when it randomly selects $\beta \in G$. Thus, CSP does not need to store DO's data to generate the proof $\{\alpha, \beta\}$ that satisfies equation (1).

4.2. The Second Type of Attack. At the TagGen stage, the CSP receives blocks and tags. CSP first calculates $s_i' = s_i \cdot P$, so it gets the following equations:

$$\begin{cases} s_1' = r_1 m_1 \cdot P + w_1 \cdot X + \phi_1 \cdot \left(D + \tau \cdot P_{\text{pub}}\right), \\ s_2' = r_2 m_2 \cdot P + w_2 \cdot X + \phi_2 \cdot \left(D + \tau \cdot P_{\text{pub}}\right), \\ \vdots \\ s_n' = r_n m_n \cdot P + w_n \cdot X + \phi_n \cdot \left(D + \tau \cdot P_{\text{pub}}\right). \end{cases} \tag{3}$$

X and D are DO's public keys, CSP knows the values of X, D, and P_{pub}, it can also calculate the value of w_i and ϕ_i for $1 \le i \le n$, and then it obtains $r_i m_i$ for $1 \le i \le n$ to calculate the following equations:

$$\begin{cases} r_1 m_1 \cdot P = s_1' - w_1 \cdot X - \phi_1 \cdot \left(D + \tau \cdot P_{\text{pub}}\right), \\ r_2 m_2 \cdot P = s_2' - w_2 \cdot X - \phi_2 \cdot \left(D + \tau \cdot P_{\text{pub}}\right), \\ r_n m_n \cdot P = s_n' - w_n \cdot X - \phi_n \cdot \left(D + \tau \cdot P_{\text{pub}}\right). \end{cases} \tag{4}$$

At the ProofGen stage, the CSP needs to calculate

$$\beta = \sum_{j\in Q} v_j \cdot m_j \cdot R_j = \sum_{j\in Q} v_j \cdot m_j r_j P. \tag{5}$$

Even if CSP deletes $\{m_1, m_2, \ldots, m_n\}$, it can calculate the value of β with $r_1 m_1 \cdot P$, $r_2 m_2 \cdot P \ldots\ldots r_n m_n \cdot P$, which can pass the audit.

5. The Improved Auditing Scheme

In this section, we first explain what an intermediate tag is and how to set an intermediate tag; then we give an improved secure auditing scheme.

We first analyze the probability of misbehavior detection in existing PDP schemes. For $n = 1000000$ 4 KB data blocks, we assume that 1% of the data blocks' integrity is damaged; TPA can specify 460 data blocks to obtain a confidence probability higher than 99%. We set n as the data blocks' total number, c_1 as damaged data blocks' number, and c_2 as randomly challenged data blocks' number during the audit. We set a random variable X representing the number of corrupted blocks in the challenged blocks; P_X represents the corresponding probability. We have the deduction as follows:

$$\begin{aligned}P_X &= P\{X \ge 1\} = 1 - P\{X = 0\} = 1 \\ &\quad - \frac{n - c_1}{n} \cdot \frac{n - 1 - c_1}{n - 1} \cdots \frac{n - c_2 + 1 - c_1}{n - c_2 + 1}.\end{aligned} \tag{6}$$

Because $(n - c_1/n) > (n - 1 - c_1/n - 1)$, so:

$$P_X \ge 1 - \left(\frac{n - c_1}{n}\right)^{c_2}. \tag{7}$$

In the case of $c_1/n = 1\%$, when c_2 is 300, 460, and 688, P_X is greater than 95%, 99%, and 99.9%, respectively. Therefore, in an audit process, few data blocks are challenged, and the relevant tags are used to generate the proof. Most of the other data blocks and relevant tags are idle.

Assuming that there are total $n = 1000000$ data blocks and tags stored in the cloud, 460 of them are challenged in each audit, and the challenged data blocks are different in multiple audits. Then it takes about 2173 audit times to use all the data blocks and corresponding tags. In practical applications, due to the user's demand for data update, many idle blocks and corresponding tags are modified and updated before they can be used, resulting in a large waste of computing overhead.

Therefore, we propose the idea of intermediate tags: at the TagGen stage, users only generate intermediate tags composed of the private key and data blocks, instead of calculating mature tags used by CSP when generating evidence, which reduces the calculation overhead of users. At the ProofGen stage, CSP calculates mature tags of only a few

challenged data blocks according to the challenge information from the TPA and uses them to generate the proof. The idea of intermediate tags is applied to the following improved scheme:

(1) Setup: KGC first selects the cyclic group G on the elliptic curve E, defines the large prime number q with the order of G, selects $P \in G$ as the generator, $H_{1,2,3,4}: \{0,1\}^* \longrightarrow Z_q^*$ as hash functions, and a random $\lambda \in Z_q^*$ as the system master key, and calculates $P_{\text{pub}} = \lambda \cdot P \in G$. Finally, KGC keeps λ in secret and exposes the public parameters.

(2) PartialKeyGen: DO sends the real identity information $\text{ID} \in Z_q^*$ to KGC. After KGC receives $\text{ID} \in Z_q^*$, it selects a random number $u \in Z_q^*$ and calculates $\text{PID}_1 = u \cdot P$, $PID_2 = I\ D \oplus H_1(u \cdot P_{pub} \| PID_1)$. Then KGC sends DO's virtual identity $PI\ D = \{PID_1, PID_2\}$ to DO and randomly selects $d \in Z_q^*$ and calculates $D = d \cdot P$, $\tau = H_2(PI\ D \| D)$, $y = d + \lambda \cdot \tau$. Finally, KGC sends DO's partial keys $\{D, y\}$ to DO.

(3) SecretValueGen: DO randomly chooses $x \in Z_q^*$ and obtains complete private key $\{x, y\}$.

(4) PublicKeyGen: DO calculates $X = x \cdot P$ and obtains the complete public key $\{D, X\}$.

(5) TagGen: the data file M is divided into n blocks by DO as $M = \{m_1, m_2, \ldots, m_n\}$, where $m_{i(1 \le i \le n)} \in Z_q^*$. DO randomly selects $r_i \in Z_q^*$, $k \in Z_q^*$ and calculates $R_i = r_i P$, $s_i = r_i m_i + k$. Note that here we have simplified the formula for calculating s_i, and the intermediate tag s_i in the improved scheme is different from the mature tag s_i in the original scheme. Thus the tags $\sigma = \{R_1, R_2, \ldots, R_n, s_1, s_2, \ldots, s_n\}$ are generated by DO; they are sent with the data blocks to CSP. DO sends k to TPA and deletes the local data and tags.

(6) Challenge: after receiving DU's audit request, TPA generates the challenge message. It first selects a random subset Q in $\{1, 2, \ldots, n\}$. The subset Q includes c elements. For $j \in Q$, TPA randomly chooses $v_j \in \mathbb{Z}_q^*$; then it sends $\text{chal} = \{j, v_j\}_{j \in Q}$ to CSP as the challenge message.

(7) ProofGen: CSP calculates $w_j = H_3(X \| R_j \| id_j)$ and $\phi_j = H_4(D \| R_j \| id_j)$ for $j \in Q$ after receiving $chal = \{j, v_j\}_{j \in Q}$. Then it calculates $\alpha = \sum_{j \in Q} v_j \cdot (s_j \cdot P + w_j X + \phi_j Y)$, $\beta = \sum_{j \in Q} v_j \cdot m_j \cdot R_j$ as the proof and sends $\{\alpha, \beta\}$ to TPA.

(8) Verify: after receiving the proof $\{\alpha, \beta\}$, TPA calculates $\tau = H_2(PI\ D \| D)$ and calculates $w_j = H_3(X \| R_j \| id_j)$, $\phi_j = H_4(D \| R_j \| id_j)$ for $j \in Q$. Then, it verifies

$$\alpha \stackrel{?}{=} \beta + \sum_{j \in Q} v_j kP + \sum_{j \in Q} w_j v_j X + \sum_{j \in Q} \phi_j v_j \left(D + \tau P_{\text{pub}}\right). \tag{8}$$

If equation (8) holds, DO's cloud data is complete. The correctness of equation (8) is as follows:

$$\begin{aligned}
\alpha &= \sum_{j \in Q} v_j \cdot \left(s_j \cdot P + w_j X + \phi_j \left(D + \tau P_{\text{pub}}\right)\right) + \sum_{j \in Q} v_j kP \\
&= \sum_{j \in Q} v_j \left(r_j \cdot m_j + w_j x + \phi_j \cdot y\right) \cdot P + \sum_{j \in Q} v_j kP \\
&= \sum_{j \in Q} v_j r_j m_j P + \sum_{j \in Q} v_j w_j x P + \sum_{j \in Q} v_j \phi_j y P + \sum_{j \in Q} v_j kP \\
&= \sum_{j \in Q} v_j m_j R_j + \left(\sum_{j \in Q} v_j w_j\right) X + \left(\sum_{j \in Q} v_j \phi_j\right) \left(D + \tau P_{\text{pub}}\right) + \sum_{j \in Q} v_j kP \\
&= \beta + \sum_{j \in Q} v_j kP + \left(\sum_{j \in Q} w_j v_j\right) X + \left(\sum_{j \in Q} \phi_j v_j\right) \left(D + \tau P_{\text{pub}}\right).
\end{aligned} \tag{9}$$

6. Analysis of the Improved Protocol

In this section, we first demonstrate that the improved scheme can resist the above attacks. Then the improved scheme's performance is analyzed. We also compare the computation overhead in two schemes, so as to prove that our improved scheme is more efficient.

6.1. Security Analysis. CSP holds the following equations in the improved scheme:

$$\begin{cases}
s_1 = r_1 m_1 + k, \\
s_2 = r_2 m_2 + k, \\
\vdots \\
s_i = r_i m_i + k.
\end{cases} \tag{10}$$

In equation (10), r_i and k are unknown to CSP; it always has more unknowns than equations, so CSP cannot solve the equations to calculate the values of r_i and k. At the ProofGen stage, CSP cannot know $r_1 m_1, r_2 m_2, \ldots, r_n m_n$. When CSP uses the second of the above attacks, it can list the following equations:

Table 2: The computational costs of the two schemes at each stage.

	The original scheme	The improved scheme
TagGen	$nM_G + 2nH + 3nM_Z + 2nA_Z$	$nM_G + nM_Z + nA_Z$
ProofGen	$(c-1)A_G + 2cM_Z + (c-1)A_Z + cM_G$	$2cH + (c-1)A_G + (5c+1)M_Z + (4c-4)A_Z + (c+4)M_G$
Verify	$(2c+1)H + 3A_G + 2cM_Z + (2c-2)A_Z + 3M_G$	$(2c+1)H + 4A_G + (3c+1)M_Z + (3c-3)A_Z + 4M_G$

$$\begin{cases} m_1 \cdot R_1 = s_1 \cdot P + k \cdot P, \\ m_2 \cdot R_2 = s_2 \cdot P + k \cdot P, \\ \vdots \\ m_i \cdot R_i = s_i \cdot P + k \cdot P. \end{cases} \tag{11}$$

Since CSP does not know the value of k, it cannot compute the value of $m_i \cdot R_i$. When generating $\beta = \sum_{j \in Q} v_j \cdot m_j \cdot R_j$, CSP can not calculate the value of β with the tag uploaded by DO. Only when the $m_1, m_2, \ldots, m_n$ are stored correctly and completely by CSP, can CSP generate the correct β and pass the TPA audit.

When CSP uses the first of the above attacks, after randomly selecting one of the values of α and β, it attempts to obtain the value of the other variable by calculating equation (1). But in equation (9), k is unknown to CSP, and CSP cannot compute β from α and equation (1) or compute α from β and equation (1).

6.2. Performance Analysis. The idea of intermediate tags is to save computing overhead for DO. The difference of storage and communication costs between two schemes is small, so we mainly analyze the computing costs of the two schemes.

In the original scheme, at the TagGen stage, DO needs to calculate $R_i = r_i P$, $w_j = H_3(X \| R_j \| id_j)$, $\phi_j = H_4(D \| R_j \| id_j)$, $s_i = r_i \cdot m_i + w_i \cdot x + \phi_i \cdot y$, and the calculation cost is $nM_G + 2nH + 3nM_Z + 2nA_Z$. At the ProofGen stage, CSP calculates $\alpha = \sum_{j \in Q} v_j \cdot s_j \cdot P$ and $\beta = \sum_{j \in Q} v_j \cdot m_j \cdot R_j$, set c as the number of elements in Q, and the calculation cost is $(c-1)A_G + 2cM_Z + (c-1)A_Z + cM_G$. At the Verify stage, TPA calculates $\tau = H_2(PI\ D \| D)$, for $j \in Q$, calculate $w_j = H_3(X \| R_j \| id_j)$, $\phi_j = H_4(D \| R_j \| id_j)$ and equation (1), the and calculation cost is $(2c+1)H + 3A_G + 2cM_Z + (2c-2)A_Z + 3M_G$.

In the improved scheme, at the TagGen stage, DO only needs to calculate $R_i = r_i P$, $s_i = r_i m_i + k$, and the computational cost is $nM_G + nM_Z + nA_Z$. At the ProofGen stage, CSP calculates $w_j = H_3(X \| R_j \| id_j)$, $\phi_j = H_4(D \| R_j \| id_j)$ for each challenged block. Then, it calculates $\alpha = \sum_{j \in Q} v_j \cdot (s_j \cdot P + w_j X + \phi_j (D + \tau P_{pub}))$, $\beta = \sum_{j \in Q} v_j \cdot m_j \cdot R_j$, and the calculation cost is $2cH + (c-1)A_G + (5c+1)M_Z + (4c-4)A_Z + (c+4)M_G$. At the Verify stage, TPA calculates $\tau = H_2(PI\ D \| D)$, calculate $w_j = H_3(X \| R_j \| id_j)$, $\phi_j = H_4(D \| R_j \| id_j)$ for $j \in Q$, and equation (1) is also calculated. The calculation cost is $(2c+1)H + 4A_G + (3c+1)M_Z + (3c-3)A_Z + 4M_G$. The computational costs of the two schemes at each stage are compared as Table 2.

As we can see from Table 2, in the improved scheme, DO reduces the computational overhead of $2nH + 2nM_Z$ at the TagGen phase. At the ProofGen phase, CSP needs to bear the extra computation overhead of $2cH + (3c+1)M_Z + (3c-3)A_Z + 4M_G$. At the Verify phase, TPA needs to bear the extra computation overhead of $A_G + (c+1)M_Z + (c-1)A_Z + M_G$. Notice that the value of n is much larger than the value of c, the extra computing overhead borne by CSP and TPA is far less than the reduced computing overhead by DO, and the improved solution is more user-friendly and more efficient.

7. Conclusion

In this paper, we point out that Ming and Shi's scheme is insecure. The aggregated data blocks required for the audit are easy to forge. CSP can provide the correct integrity proof after modifying or deleting the data, and TPA will give the correct integrity audit results. In addition, to solve the idle tags problem in the existing audit schemes, we propose the idea of intermediate tags, which can save computing power for users. Finally, we apply the idea to the improved scheme and upgrade the original scheme on security to solve the security problems of the Ming and Shi's scheme and improve the audit efficiency. We hope that our idea of intermediate tags can be used by more scholars to construct more efficient audit solutions and the security issue pointed by us can be avoided when they design the scheme.

Data Availability

The datasets of this article are available on request from the authors.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Xiuguang Li and Ruifeng Li are responsible for the writing of the article and the construction of the improved scheme, Xu An Wang is responsible for the derivation of the formulas in the article and gives some significant ideas, Ke Niu is responsible for the verification of the security of this article, Xiaoyuan Yang is responsible for the polishing of the language of the article and the collection of the information related to this article, and Hui Li revised the finished manuscript.

Acknowledgments

This work was supported by National Key Research and Development Program of China (no. 2017YFB0802000); National Natural Science Foundation of China (nos. 62172436, 62102452, and 61732022); National Natural Science Foundation of China Key Program (U1836203); State Key Laboratory of Public Big Data (no. 2019BDKFJJ008); Engineering University of PAP's Funding for Scientific Research Innovation Team (no. KYTD201805); and Engineering University of PAP's Funding for Key Researcher (no. KYGG202011).

References

[1] L. Song, Y. Miao, J. Weng, K.-K. R. Choo, X. Liu, and R. H. Deng, "Privacy-Preserving threshold-based image retrieval in cloud-assisted internet of things," *IEEE Internet of Things Journal*, vol. 20229 pages, Article ID 3142933, 2022.

[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*, pp. 598–609, Association for Computing Machinery, Alexandria, VA, USA, October 2007.

[3] Z. Ma, J. Ma, Y. Miao et al., "Lightweight privacy-preserving medical diagnosis in edge computing," in *Proceedings of the 2021 IEEE World Congress on Services (SERVICES)*, p. 9, IEEE, Chicago, IL, USA, September 2021.

[4] Y. Ming and W. Shi, "Efficient privacy-preserving certificateless provable data possession scheme for cloud storage," *IEEE Access*, vol. 7, Article ID 122091, 2019.

[5] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[6] H. Jin, H. Jiang, and K. Zhou, "Dynamic and public auditing with fair arbitration for cloud data," *IEEE Transactions on Cloud Computing*, vol. 6, no. 3, pp. 680–693, 2018.

[7] H. Tian, Y. Chen, C. C. Chang et al., "Dynamic-hash-table based public auditing for secure cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 701–714, 2017.

[8] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2402–2415, 2017.

[9] W. Guo, H. Zhang, S. Qin et al., "Outsourced dynamic provable data possession with batch update for secure cloud storage," *Future Generation Computer Systems*, vol. 95, pp. 309–322, 2019.

[10] G. Hou, J. Ma, C. Liang, and J. Li, "Efficient audit protocol supporting virtual nodes in cloud storage," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 5, pp. 1–14, 2020.

[11] R. Mishra, D. Ramesh, and D. R. Edla, "BB-tree based secure and dynamic public auditing convergence for cloud storage," *The Journal of Supercomputing*, vol. 77, no. 5, pp. 4917–4956, 2021.

[12] 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, IEEE, Beijing, China, June 2008.

[13] C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, and J. Chen, "MuR-DPA: top-down levelled multi-replica Merkle hash tree based secure public auditing for dynamic big data storage on cloud," *IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2609–2622, 2015.

[14] W. Guo, S. Qin, F. Gao et al., "Dynamic proof of data possession and replication with tree sharing and batch verification in the cloud," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 1813–1824, 2022.

[15] Z. Yaling and S. Li, "Dynamic flexible multiple-replica provable data possession in cloud," in *Proceedings of the 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 291–294, IEEE, Chengdu, China, July 2020.

[16] Y. Qi, X. Tang, and Y. Huang, "Enabling efficient verification of dynamic data possession and batch updating in cloud storage," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 6, pp. 2429–2449, 2018.

[17] K. Deng, M. Xu, and S. Fu, "Outsourced data integrity auditing for efficient batch dynamic updates," *Communications in Computer and Information Science*, vol. 1149, pp. 325–339, 2020.

[18] Z. Ma, J. Ma, Y. Miao et al., "Verifiable data mining against malicious adversaries in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 953–964, 2022.

[19] X. Wang, J. Ma, Y. Miao, X. Liu, and R. Yang, "Privacy-Preserving diverse keyword search and online pre-diagnosis in cloud computing," *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 710–723, 2022.

[20] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, and K.-K. R. Choo, "Fuzzy identity-based data integrity auditing for reliable cloud storage systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 1, pp. 72–83, 2019.

[21] B. Wang, B. Li, and H. Li, "Oruta: privacy-preserving public auditing for shared data in the cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 1, pp. 43–56, 2014.

[22] X. Wang, W. Jiao, H. Yang, L. Guo, X. Ye, and Y. Guo, "Algebraic signature based data possession checking method with cloud storage," in *Proceedings of the 11th International Conference on Prognostics and System Health Management*, pp. 11–16, IEEE, Jinan, China, October 2020.

[23] F. Li, J. Ma, Y. Miao et al., "Towards efficient verifiable boolean search over encrypted cloud data," *IEEE Transactions on Cloud Computing*, vol. 2021, Article ID 3118692, 1 page, 2021.

[24] Y. Zhang, C. Xu, S. Yu, H. Li, and X. Zhang, "SCLPV: secure certificateless public verification for cloud-based cyber-physical-social systems against malicious auditors," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 4, pp. 159–170, 2015.

[25] J. Li, J. Wu, G. Jiang, and T. Srikanthan, "Blockchain-based public auditing for big data in cloud storage," *Information Processing & Management*, vol. 57, no. 6, Article ID 102382, 2020.

[26] H. Yang, R. Su, P. Huang et al., "PMAB: a public mutual audit blockchain for outsourced data in cloud storage," *Security and Communication Networks*, vol. 202111 pages, Article ID 9993855, 2021.

[27] X. Yang, X. Pei, M. Wang, T. Li, and C. Wang, "Multi-replica and multi-cloud data public audit scheme based on blockchain," *IEEE Access*, vol. 8, Article ID 144809, 2020.

[28] 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.

[29] P. Wei, D. Wang, Y. Zhao, S. K. S. Tyagi, and N. Kumar, "Blockchain data-based cloud data integrity protection mechanism," *Future Generation Computer Systems*, vol. 102, pp. 902–911, 2020.

[30] I. Zikratov, A. Kuzmin, V. Akimenko, V. Niculichev, and L. Yalansky, "Ensuring data integrity using blockchain technology," in *Proceedings of the 20th Conference of Open Innovations Association (FRUCT)*, pp. 534–539, IEEE, Saint-Petersburg, Russia, April 2017.

[31] W. Xu, D. Feng, and J. Liu, "Public verifiable proof of storage protocol from lattice assumption," in *Proceedings of the 2012 IEEE International Conference on Intelligent Control, Automatic Detection and High-End Equipment*, pp. 133–137, IEEE, Beijing, China, July 2012.

[32] H. Liu and W. Cao, "Public proof of cloud storage from lattice assumption," *Chinese Journal of Electronics*, vol. 23, no. 1, pp. 186–190, 2014.

[33] X. Zhang, C. Xu, and C. Jin, "Enabling identity-based cloud storage public auditing with quantum computers resistance," *International Journal of Electronic Security and Digital Forensics*, vol. 8, no. 1, pp. 82–98, 2016.

[34] X. Zhang, H. Wang, and C. Xu, "Identity-based key-exposure resilient cloud storage public auditing scheme from lattices," *Information Sciences*, vol. 472, pp. 223–234, 2019.

[35] C. Sasikala and C. Shoba Bindu, "Certificateless remote data integrity checking using lattices in cloud storage," *Neural Computing & Applications*, vol. 31, no. 5, pp. 1513–1519, 2019.

[36] C. Lan, H. Li, and C. Wang, "Cryptanalysis of "Certificateless remote data integrity checking using lattices in cloud storage," in *Proceedings of the 10th International Conference on Information Science and Technology (ICIST)*, pp. 134–138, IEEE, London, UK, September 2020.