Hindawi
Security and Communication Networks
Volume 2021, Article ID 4485311, 12 pages
https://doi.org/10.1155/2021/4485311

WILEY | Hindawi

Research Article

Privacy-Preserving Redactable Blockchain for Internet of Things

Yanli Ren, Xianji Cai, and Mingqi Hu

School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China

Correspondence should be addressed to Yanli Ren; renyanli@shu.edu.cn

Received 8 June 2021; Revised 4 August 2021; Accepted 25 August 2021; Published 20 September 2021

Academic Editor: Jinguang Han

In the traditional blockchain system, data is public and cannot be redacted. With the development of blockchain technology, the problem that the data cannot be altered will be more serious once it is written on the chain. Recently, some redactable blockchain schemes have been proposed. However, most of the schemes are based on the public blockchain, and the users' identities and transaction data may be disclosed. To solve the problem of privacy disclosure, we propose a privacy-preserving transaction-level redactable blockchain. In the proposed scheme, symmetric encryption and ring signature are used to protect transaction data and the users' identities, respectively. In order to prove the legality of data redaction, the transaction sender can reveal the invalid users' identities and transaction data in an anonymous environment. To construct a transaction-level redactable blockchain, the users only need to replace a single transaction to complete the data redaction instead of replacing the entire block. The experimental results show that the proposed scheme saves 20% of the redaction time compared to the previous privacy-preserving blockchains, so the redaction efficiency is higher.

1. Introduction

In January 2009, blockchain was invented as the underlying technology of Bitcoin [1], which merged the important achievements in some fields such as modern cryptography and distributed network. After the emergence of Bitcoin, the blockchain network has supported massive transfer transactions steadily in a purely distributed manner [2]. Especially, the combinations of blockchain technology and the Internet of things (IoT) begin to appear in large numbers, which proves that the blockchain is a good solution to the basic needs of distributed networks [3, 4]. For example, the blockchain is used to ensure the safety and anticounterfeiting traceability of the food supply chain [5].

With the wide application of blockchain technology in E-government, military, and medical fields [6–8], the problem that the data cannot be altered will be more serious. The continuous growth of data in the blockchain leads to a dramatical increase in storage space and data verification cost, which will result in the continuous decline of the number of the full-nodes and intensify the trend of centralization in blockchain. Moreover, the computational overhead for the nodes to verify historical data will also increase, which is unfriendly to some IoT devices with limited computing resource [9]. Therefore, redacting or deleting the useless data will be an important means to improve the performance and scalability of the blockchain.

Recently, some redactable blockchains have been proposed [10, 11]. However, the users' identities and transactions are public. Even if the IoT devices simply apply pseudonyms when acting in the blockchain, they can only provide limited identity privacy. The adversary can still trace the transparent block data to find out the relations between the identities and transactions and analyze the real identities of the users [12]. Moreover, in the process of data redaction, the personal information of the redactors may be revealed. Due to the wide applications of IoT devices, the data owners do not want to leak the identity privacy or business secrets to their competitors. Therefore, it is extremely important to propose a privacy-preserving redactable blockchain to simultaneously protect the privacy of the users and rectify the incorrect data.

1.1. Privacy-Preserving Blockchain. An important feature of blockchain is data transparency, which means that the

identity and transaction data in public blockchain are transparent. However, for some companies or users, data is extremely important to maintain their strong competitiveness, and sharing data may bring some security challenges. Therefore, more and more privacy-preserving mechanisms on blockchain have been proposed by researchers. At present, the most popular solutions to protect the identity privacy are ring signature [13], mixing services [14], and noninteractive zero-knowledge proof [15], and the most common methods to protect the data privacy include zero-knowledge proof, homomorphic encryption, and commitment schemes [16].

1.2. Redactable Blockchain. In [10], the concept of redactable blockchain is first raised. The main idea of their protocol was to keep the block hash link compatible with the current state when a redaction happens. They use a special hash function called chameleon hash [17] to replace the traditional collision-resistant hash function, where collisions of the hash can be computed with trapdoors. Thus, a collision can be calculated to keep the hash linked as usual. In this work, the authors improve the chameleon hash protocol and solve the key exposure problem in the previous chameleon hash and also provide the formal security analysis of their new protocol. However, in this scheme, a trusted-party is needed to control the trapdoor of chameleon hash, which violates the idea of decentralization of the blockchain, and the redaction operation does not represent the opinions of the whole system. Moreover, this scheme can only realize data redaction at the block-level. Once a data is redacted, the whole block needs to be replaced.

Later, Deuber et al. proposed a history dependent redactable scheme [11], which is dependent on the history of the transaction data. They propose a redactable protocol in the permissionless setting, which can be integrated to Bitcoin easily. The scheme preserves the original transaction merkle root to keep the hash linked if a block is altered. In their scheme, complex cryptographic tools or trusted parties are not used, and the experiment proves that their scheme is an efficient and feasible protocol. However, in this scheme, consensus-based voting is used during the process of redaction, and this will lead to the privacy disclosure of the redactor. Moreover, this scheme also realizes data redaction at the block-level, but not on the transaction-level. Once a data is redacted, the whole block needs to be replaced.

Most of the redactable blockchains are realized on the block-level, which means that transactions can only be redacted by a block as a unit. Recently, a transaction-level redactable blockchain was proposed [18], where only the hash collision of the redacted transaction should be found. They propose a policy-based chameleon hash, which combines the ciphertext-policy attributed based encryption (CP-ABE) algorithm with a chameleon hash scheme. In their scheme, the attributes of a user should satisfy the access structure of the CP-ABE algorithm [18]. The chameleon hash collision can only be found when a user obtains the short-term and the long-term trapdoors at the same time.

However, all of the solutions to blockchain redaction ignore the issue of data privacy disclosure. Recently, a deletable blockchain was proposed \cite{cai2019privacy}, where the identities and transactions of the users are well protected when a block is deleted. In order to delete a block in an anonymous environment, the real identities or the transaction data are disclosed through traceable ring signature or Petersen commitment. Moreover, a linkable multisignature scheme was proposed to prevent the disclosure of the users' identities, which can trace an adversary if he generates a signature more than one time. However, the block can only be deleted by the sender of a transaction but cannot be redacted. Thus, it is urgent and meaningful to propose a redactable blockchain without disclosing the identities and transaction data of the users.

1.3. Our Contributions

(1) We propose a privacy-preserving redactable blockchain protocol without disclosing the identities and transaction data of the users. A threshold ring signature and a symmetric encryption algorithm are separately used to protect the users' identities and transaction data.

(2) The proposed blockchain is a fine-grained transaction-level redactable one. The block data can be redacted on the transaction level instead of block level, which means that other transactions in the block do not need to be changed when one transaction is redacted.

(3) We formalize four security requirements for a privacy-preserving redactable blockchain including identity privacy, data privacy, chain quality, and common prefix and prove that our proposed protocol is provably secure based on symmetric encryption and threshold ring signature.

(4) The efficiency of the proposed protocol is demonstrated by a series of experiments. The result shows that the time of redacting a transaction is about twice that of creating a block, and the proposed protocol can save about 20% of computational costs than the previous ones.

2. Preliminaries

Some definitions and algorithms utilized in the proposed protocol are introduced in this section.

2.1. An Introduction of Blockchain. Assume that B_{j-1}, B_j, B_{j+1} are three adjacent blocks in a blockchain. As shown in Figure 1, the j-th block is denoted as $B_j = \langle s_j, x_j, \mathrm{ctr}_j \rangle$. The block can point to its previous block through the hash value $s_j \in \{0\ ,\ 1\}^k$. All of the transactions packaged in a block are contained in $x_j \in \{0\ ,\ 1\}^*$, and $\mathrm{ctr} \in \mathbb{N}$ is the result of the PoW puzzle [20]. The users check whether the block B_j is valid or not by the following inequality:

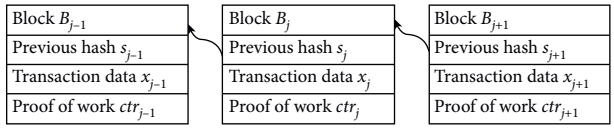

Figure 1: Block structure in a blockchain.

$$\text{validateBlockD}\left(B_j\right) = H_1\left(\text{ctr}_j, H_2\left(s_j, x_j\right)\right) < D, \quad (1)$$

where the parameter D is the target difficulty coefficient of the PoW consensus mechanism. H_1, H_2: $\{0,1\}^* \longrightarrow \{0,1\}^k$ are two different cryptographic hash functions.

A blockchain is composed of a series of blocks, which have solved the PoW puzzle. The newly created block points to its previous block through the hash value. We denote the length of a chain $\text{len}(\mathscr{C})$, and the rightmost block of chain $\mathscr{C}$ can be denoted as $\text{len}(\mathscr{C}) = \langle x', s', \text{ctr}' \rangle$. It should be noted that a blockchain can be an empty chain, which is denoted as $\mathscr{C} = \mathscr{E}$.

The chain $\mathscr{C}$ will be updated as $\mathscr{C}' = \mathscr{C} \| B_{j+1}$ if a new block B_{j+1} is generated and validated by the users, and at present, the last block of the chain $\mathscr{C}'$ is the block B_{j+1}. For any $q \geq 0$, $\mathscr{C}^q$ is denoted as the chain after deleting the rightmost q blocks. Note that $\mathscr{C}^q$ can be an empty chain if q is larger than n, where n is the length of a chain $\mathscr{C}$. If the prefix of chain $\mathscr{C}_1$ is $\mathscr{C}_2$, we write $\mathscr{C}_1 \preceq \mathscr{C}_2$.

2.2. Redactable Protocol in the Public Blockchain. In this subsection, we introduce a redactable blockchain protocol [11], which will be used in our paper. The main contribution is that it preserves the old state $o_j = H_2(s_j, x_j)$ of a block, which is actually the merkle root of the old transactions. As shown in Figure 2, a block is linked to its predecessor by two links, an old hash link (solid arrow) and a new hash link (dashed arrow), and $s_{j+1} = H_1(\text{ctr}_j, H_2(s_j, x_j), o_j)$ holds [11]. When the block B_j is redacted, the new hash link between the block B_j and B_{j+1} is broken (marked by a red cross) because $s_{j+1} \neq H_1(\text{ctr}_j, H_2(s_j, x_j^*), o_j)$. However, B_{j+1} can still point to B_j successfully through the old hash link (solid arrow) since the merkle root of the old transaction (old state) is still preserved. The reason is shown as follows:

$$\begin{aligned} s_{j+1} &= H_1\left(\text{ctr}_j, o_j, o_j\right) \\ &= H_1\left(\text{ctr}_j, H_2\left(s_j, x_j\right), o_j\right). \end{aligned} \quad (2)$$

Please see reference [11] for the details.

2.3. Threshold Ring Signature. The following algorithms are commonly included in a (t, n) threshold ring signature (TRS) scheme, where t $(0 \leq t \leq n)$ is the value of the threshold, n is the number of the ring members, and $t < n$ [21]:

TRS_Setup: on input security parameter λ, it outputs the public keys $pk_1, \ldots, pk_n$ and private keys $sk_1, \ldots, sk_n$ of n ring members

TRS_Sign: on input a message $m \in \{0,1\}^*$, a ring containing n members with n public keys $pk_1, \ldots, pk_n$, and private keys $sk_1, \ldots, sk_n$ of t members, it outputs a threshold ring signature σ on the message m

TRS_Verify: on input a message m, a signature σ and public keys $pk_1, \ldots, pk_n$ of n members in a ring, it outputs 0 or 1 to indicate accepting or rejecting the signature

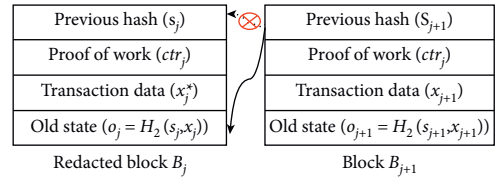

Figure 2: The redacted block can still be linked by the old state.

3. Privacy-Preserving Redactable Blockchain

This section describes a privacy-preserving redactable blockchain protocol Γ^*, which can redact the block data on the transaction-level without disclosing the transaction data and the identities of the users.

3.1. Syntax of Privacy-Preserving Redactable Blockchain. We construct a privacy-preserving redactable blockchain protocol Γ', which contains the following four algorithms:

$\{0,1\} \longleftarrow \Gamma'.\text{validateReq}(\mathscr{C}, j, i, tx_i^*, m, K)$: on inputting a chain $\mathscr{C}$, an index j of the redacted block, an index i of the redacted transaction, an initial candidate transaction tx_i^*, the old transaction data m, and the old encryption key K, it returns 1 if the redaction request is valid; otherwise, it returns 0.

$\{0,1\} \longleftarrow \Gamma'.\text{validateCandTx}(\mathscr{C}, tx_i{}^{**}, j)$: on inputting a chain $\mathscr{C}$, a final candidate transaction $tx_i{}^{**}$ and an index j, it returns 0 if $tx_i{}^{**}$ is valid; otherwise, it returns 1.

$\{0,1\} \longleftarrow \Gamma'.\text{validateChain}(\mathscr{C})$: on inputting a chain $\mathscr{C}$, it returns 1 if the chain is valid; otherwise, it returns 0.

$\{0,1\} \longleftarrow \Gamma'.\text{validateBlock}(B)$: on inputting a block B, it returns 1 if the block is valid; otherwise, it returns 0.

3.2. Our Proposed Protocol. In the proposed protocol, a regular transaction is denoted as $tx = \langle E_K(m), \text{RSig} \rangle$, where m is the transaction data including the input and output of a transaction, $E(\cdot)$ is a symmetric encryption algorithm, $E_K(m)$ is the ciphertext of the transaction data using a key K and RSig is a ring signature to protect the identity privacy of the users. H: $\{0,1\}^* \longrightarrow \{0,1\}^k$ is a cryptographic hash function. The block data is denoted as $x = \{tx_1, tx_2, \ldots, tx_p\}$, where p is the number of the transactions packaged in each block.

3.2.1. Proposing a Redaction Request. Once a user joins in the blockchain system, he needs to generate a public key as his identity, and a private key, which will be used to generate a signature. If a miner wants to participate in a redaction

process, he can apply for the key generation algorithm of the threshold ring signature system to get another key pair when he joins the blockchain system. The key generation algorithm is executed by a trusted party, such as certification authority (CA). It should be noted that the private key generated by the user as the identity is different from that generated by the TRS system, where the former is used to generate the ring signature RSig to protect the identity of the users, and the latter is used to generate the threshold ring signature TRS to allow redacting a transaction in a block.

If a user wants to propose a redaction request for the transaction tx_i in block B_j, he discloses the transaction encryption key K and the encrypted transaction data m first. Then, he generates a new transaction $m\prime$, picks up a new encryption key $K\prime$, and calculates $H(m, K\prime, m\prime, K)$, $E_{K\prime}(m\prime)$, ring signature RSig$\prime$ and a ring signature S on $H(m, K\prime, m\prime, K)$. Next, the user generates an initial candidate transaction $tx_i^* = \langle E_{K\prime}(m\prime), \text{RSig}\prime, H(m, K\prime, m\prime, K), S\rangle$. Finally, the user broadcasts the redaction request $r_{j,i} = \langle j, i, tx_i^*, m, K\rangle$ to the whole network. Note that two ring signatures are both generated by the data owner, who randomly chooses two rings including some users on the chain, and compute the signatures by the public keys of these users.

3.2.2. Joining a Redaction Operation. When a miner in the blockchain receives a redaction request, he invokes Algorithm 1 Γ'.validateReq to judge whether the redaction request is valid or not. If the redaction request is valid, he votes for the redaction by putting his vote result in the latest block he mines. If the number of miners who approve of the redaction request exceeds the threshold t of the threshold ring signature, these miners generate a threshold ring signature TRS jointly on the redaction message. Finally, a final candidate transaction tx_i^{**} is generated and broadcasted to the network, where $tx_i^{**} = tx_i^* \| \text{TRS}$.

3.2.3. Validating a Redaction Operation. Everyone can verify whether a final candidate transaction is valid or not by invoking Γ'.validateCandTx (Algorithm 2). If the final candidate transaction tx_i^{**} is valid, the user replaces the old transaction tx_i in block B_j with the final candidate transaction tx_i^{**} in their local chain. The process of redacting a transaction is shown in Figure 3.

4. Protocol Description

In this subsection, we introduce each algorithm in the proposed protocol in detail.

Algorithm 1 can validate whether a redaction request is valid or not. The miners generate a threshold ring signature if the validation of step 5 to step 9 all passed.

Algorithm 2 can validate whether a final candidate transaction is valid or not. This algorithm first checks whether the redacted block B_j^* is valid or not. In line 6, Algorithm 2 checks the validity of the threshold ring signature TRS. In line 11, it checks the old hash link from B_j^* to B_{j-1} and the hash link from B_{j+1} to B_j^*. The final candidate transaction can be considered valid if the hash link of block B_j can point to the previous block correctly. The user can replace the old transaction with the final candidate transaction in his local chain.

Algorithm 3 describes how to validate a chain $\mathscr{C}$ in our system. This algorithm checks the chain $\mathscr{C}$ from the beginning to the end. Users only have to validate the head of the chain $\mathscr{C}$ if the length of a chain is 1. Otherwise, blocks should be validated one by one. In line 8, the users first check the current link between the neighboring blocks; if the link is broken, then the users check the old link in line 10. The users accept the block if the old link can point to the previous block successfully, and the candidate block is valid.

Algorithm 4 describes how to validate a block B in our system. All of the transactions in the block should be checked using some predefined validation rules. In line 2, this algorithm checks whether the block B has solved the PoW puzzle or not. If the block has been redacted before, the old state of B should be checked to judge whether the block has solved the PoW puzzle or not.

5. Security Analysis

The following contents provide security analysis of our privacy-preserving redactable blockchain protocol Γ'. The original public redactable blockchain protocol Γ satisfies two security requirements: chain quality and common prefix. Supposing that the protocol is a secure and reliable system, the proposed protocol Γ' also satisfies these two requirements. Most importantly, Γ' can protect the users' identities and transaction data simultaneously. The following description will clearly illustrate that the proposed protocol is provably secure based on threshold ring signature, symmetric encryption, and the blockchain protocol proposed in [11].

5.1. Identity Privacy. The subsection states that the users' identities are private in our protocol.

Theorem 1. *The redactable blockchain scheme realizes the privacy of the users' identities if the threshold ring signature is anonymous.*

Proof. Proof. The game is executed between an adversary $\mathscr{A}$ and a simulator $\mathscr{B}$. At the beginning of the game, $\mathscr{B}$ is given a tuple $(Y, V_0, V_1, M, \text{TRS}(M, Y, V_b))$, where $b \in \{0, 1\}$, V_0, V_1 are both subsets of Y with same size, M is the redaction message including the redaction reasons and timestamp, and $\text{TRS}(M, Y, V_b)$ is the threshold ring signature generated by V_b. Assuming that adversary $\mathscr{A}$ obtains the users' identities with a nonnegligible probability in the proposed protocol, simulator $\mathscr{B}$ can obtain the identities of the real signers with a nonnegligible probability in the threshold ring signature:

Setup. The simulator sets up the blockchain system and sends the public parameters $(\text{TRS}(\cdot), E(\cdot), H(\cdot))$ to the adversary, where $\text{TRS}(\cdot)$ is a threshold ring signature

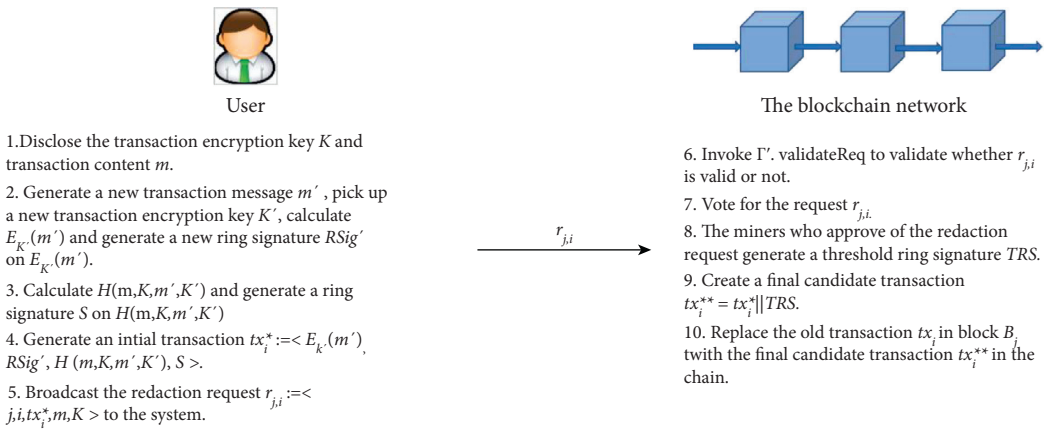

Figure 3: Proposing a redaction for the transaction tx_i of block B_j.

Input: Blockchain $\mathscr{C}$, redaction request $\mathbf{r_{j,i}} = \langle \mathbf{j}, \mathbf{i}, \mathbf{tx_i^*}, \mathbf{m}, \mathbf{K} \rangle$
Output: 0/1
(1) Parse the original **j** -th block $\mathbf{B_j^*} = \langle \mathbf{s_j}, \mathbf{x_j^*}, \mathbf{ctr_j}, \mathbf{o_j} \rangle$;
(2) Parse the old block data $\mathbf{x_j} = \{\mathbf{tx_1}, \mathbf{tx_2}, \ldots, \mathbf{tx_p}\}$;
(3) Replace the old transaction $\mathbf{tx_i}$ with the initial candidate transaction $\mathbf{tx_i^*}$;
(4) Parse the new block data $\mathbf{x_j^*} = \{\mathbf{tx_1}, \mathbf{tx_2}, \ldots, \mathbf{tx_i^*}, \ldots, \mathbf{tx_p}\}$;
(5) Parse the redacted **j**-th block $\mathbf{B_j^*} = \langle \mathbf{s_j}, \mathbf{x_j^*}, \mathbf{ctr_j}, \mathbf{o_j} \rangle$;
(6) If **vali da teBlock**$(\mathbf{B_j^*}) = 0$, return 0;
(7) Parse the initial candidate transaction $\mathbf{tx_i^*} = \langle \mathbf{E_{K\prime}(m\prime)}, \mathbf{RSig\prime}, \mathbf{H(m, K, m', K')}, \mathbf{S} \rangle$.
(8) Calculate $\mathbf{E_K(m)}$ using the disclosed transaction data **m** and encryption key **K**.
(9) Compare $\mathbf{E_K(m)}$ with the ciphertext in the original transaction. If not equal, return 0;
(10) Validate the ring signature **S** contained in the redacted transaction $\mathbf{tx_i^*}$ and compare the message signed by **S** with the digest $\mathbf{H(m, K, m', K')}$, if not equal return 0;
(11) Return 1.

Algorithm 1: **vali da teReq** (implements Γ′.**vali da teReq**).

Input : Chain $\mathscr{C}$, a final candidate transaction $\mathbf{tx_i^{**}}$, an index **j** of the block
Output : 0/1
(1) Parse $\mathbf{x_j^*} = \{\mathbf{tx_1}, \mathbf{tx_2}, \ldots, \mathbf{tx_i^*}, \ldots, \mathbf{tx_p}\}$;
(2) Parse $\mathbf{B_j^*} = \langle \mathbf{s_j}, \mathbf{x_j^*}, \mathbf{ctr_j}, \mathbf{o_j} \rangle$;
(3) Parse $\mathbf{tx_i^{**}} = \mathbf{tx_i^*} \| \mathbf{TRS}$.
(4) if **vali da teBlock**$(\mathbf{B_j^*}) = 0$ then
(5) return 0;
(6) if **TRS** is invalid then
(7) return 0;
(8) Compare the message signed by **TRS** and the hash of the redaction request, If not equal, return 0;
(9) Parse $\mathbf{B_{j-1}} = \langle \mathbf{s_{j-1}}, \mathbf{x_{j-1}}, \mathbf{ctr_{j-1}}, \mathbf{o_{j-1}} \rangle$;
(10) Parse $\mathbf{B_{j+1}} = \langle \mathbf{s_{j+1}}, \mathbf{x_{j+1}}, \mathbf{ctr_{j+1}}, \mathbf{o_{j+1}} \rangle$;
(11) if $\mathbf{s_j^*} = \mathbf{H(ctr_{j-1}, o_{j-1}, o_{j-1})} \wedge \mathbf{s_{j+1}} = \mathbf{H(ctr_j, o_j, o_j)}$ then
(12) return 1;
(13) else
(14) return 0;

Algorithm 2: **vali da teCan dT x**(implements Γ′.**vali da teCan dT x**).

```
Input: Chain 𝒞 = (B_1, B_2, ..., B_n) of length n.
Output: 0/1
(1) j: = n;
(2) if j = 1 then
(3) return vali da teBlock(B_1);
(4) while j ≥ 2 do
(5) Parse B_j = ⟨s_j, x_j, ctr_j, o_j⟩;
(6) if vali da teBlock(B_j) = 0 then
(7) return 0
(8) if s_j = H_1(ctr_{j−1}, H_2(s_{j−1}, x_{i−1}), o_{j−1}) then
(9) j = j + 1;
(10) else if s_j = H_1(ctr_{j−1}, o_{j−1}, o_{j−1}) then
(11) j = j + 1;
(12) else
(13) return 0;
(14) return 1;
(15) return 1
```

Algorithm 3: **vali da teChain**(implements Γ′.**vali da teChain**).

```
Input: Block B = ⟨s, x, ctr, o⟩
Output: 0/1
(1) if all the transactions in x are valid then
(2) if H_1(ctr, H_2(s, x), o) < D ∨ H_1(ctr, o, o) < D then
(3) return 1;
(4) else
(5) return 0;
(6) else
(7) return 0;
```

Algorithm 4: **vali da teBlock**(implements Γ.**vali da teBlock**).

scheme, $E(\cdot)$ is a symmetric encryption algorithm, and $H(\cdot)$ is a secure hash function.

Oracle Queries. The adversary simulates the following oracles:

Transaction Generation Oracle. Adversary $\mathcal{A}$ sends an index j, some transactions collected from the blockchain system. The simulator $\mathcal{B}$ returns a block $tx_i := \langle E_K(m_i), \mathrm{RSig}_i \rangle$.

Transaction Redaction Oracle. Adversary $\mathcal{A}$ sends a block index j, a transaction index i, an initial candidate transaction tx_i^*, an old transaction message m and an old transaction encryption key K, a redaction message M and two sets Y and V to the simulator, where Y is a public key set, V is a subset of Y and $V \neq V_0$ or V_1. The simulator returns the corresponding final candidate transaction $tx_i^{**} = tx_i^* \| \mathrm{TRS}(M, Y, V)$.

Challenge. The simulator $\mathcal{B}$ generates the final candidate transaction $\widehat{tx_i^*} = \widehat{tx_i^*} \| \mathrm{TRS}(M, Y, V_b)$, where M contains the disclosed transaction data m and encryption key K included in transaction $\widehat{tx_i^*}$. Then, he transmits $\widehat{tx_i^{**}}$ to the adversary $\mathcal{A}$. $\mathcal{A}$ outputs $b\prime \in \{0, 1\}$ and then $\mathcal{B}$ submits $b = b\prime$. Assuming that $\mathcal{A}$ can guess the users' identities successfully with a probability of more than 1/2, the simulator $\mathcal{B}$ can distinguish the identities of the real signers with a probability of more than 1/2, and it contradicts the anonymity of the threshold ring signature. Therefore, it is impossible for the adversary $\mathcal{A}$ to guess the identities of the signers in a threshold ring signature successfully with a probability of more than 1/2, and the proposed redactable protocol can protect the users' identities effectively. □

5.2. Data Privacy. The subsection states that the transaction data in our blockchain system is private.

Theorem 2. *The redactable blockchain scheme realizes the privacy of transaction data if the symmetric encryption algorithm is indistinguishable against chosen plaintext attack (IND-CPA) secure.*

Proof. The game is executed between an adversary $\mathcal{A}$ and a simulator $\mathcal{B}$. Assuming that adversary $\mathcal{A}$ can guess the transaction data with a nonnegligible probability in the proposed protocol, the simulator $\mathcal{B}$ can distinguish the plaintext with a nonnegligible probability in the symmetric encryption scheme:

Setup. The simulator sets up the blockchain system and generates a symmetric encryption algorithm E and an encryption key K and then sends the public parameters E to the adversary. The encryption key K is kept secret itself.

Oracle Queries. The adversary simulates the following oracles:

Encryption Oracle. The adversary sends the plaintext of transaction data m to the simulator. The simulator returns the corresponding ciphertext $tx := \langle E_K(m), \mathrm{RSig}\rangle$, where $E_K(m)$ is the ciphertext of m by using the symmetric encryption algorithm, and RSig is the ring signature of the simulator on the ciphertext $E_K(m)$.

Challenge. Adversary $\mathscr{A}$ sends a tuple $(m_0, m_1, K_0, K_1, E_{K_b}(m_b))$ to the simulator. Then, $\mathscr{B}$ randomly chooses $b \in \{0,1\}$ and computes $E_{K_b}(m_b)$. Note that $E_{K_b}(m_b)$ has never been queried in the Encryption oracles. Next, the simulator $\mathscr{B}$ generates the ring signature RSig^* on the ciphertext $E_{K_b}(m_b)$ and transmits the transaction ciphertext $tx^* := \langle E_{K_b}(m_b), \mathrm{RSig}^*\rangle$ to adversary $\mathscr{A}$.

$\mathscr{A}$ outputs $b\prime \in \{0,1\}$ and then $\mathscr{B}$ submits $b = b\prime$. Assuming that $\mathscr{A}$ can guess the transaction data successfully with a probability of more than 1/2, the simulator $\mathscr{B}$ can distinguish the ciphertext of two plaintexts with a probability of more than 1/2, and it contradicts the IND-CPA security of the symmetric encryption schemes. Therefore, it is impossible for the adversary $\mathscr{A}$ to guess the plaintext of transaction data successfully with a probability of more than 1/2, and the proposed redactable protocol can protect the transaction data effectively. □

5.3. Chain Quality. This property points out that the proportion of adversary blocks in a blockchain is limited by the computing capabilities held by the adversaries.

Definition 1. The property of (μ, ℓ) chain quality is parameterized by $\mu \in \mathbb{R}$ and $\ell \in \mathbb{N}$, which states that, for any part of the chain composed by ℓ continuous blocks held by the honest party, the proportion of adversarial blocks is at most μ, where μ is used to evaluate the computing capabilities of the adversaries.

Theorem 3. *Γ' satisfies (μ, ℓ)-chain quality if the public immutable blockchain satisfies (μ, ℓ)-chain quality for any (t, n) threshold ring signature where $t/n > \mu$.*

Proof. Proof. Different from the public immutable blockchain, we provide the function of transaction redaction in an anonymous environment. In order to increase the proportion of malicious blocks in the chain and break the chain quality property, adversary $\mathscr{A}$ could redact an honest transaction tx_i of block B_j into a malicious transaction tx_i^{**}. We prove that the probability of replacing an honest transaction with a malicious one by the adversary $\mathscr{A}$ is negligible.

Suppose that adversary $\mathscr{A}$ proposes an initial candidate transaction tx_i^* to an honest transaction tx_i of block B_j. Limited by computing resources, the adversary $\mathscr{A}$ can only mine μ ratio of blocks during the phase of threshold ring signature generation. However, the proposed scheme requires that the ratio of redaction supporters to generate the threshold ring signature has to be at least t/n, where $t/n > \mu$. The adversary $\mathscr{A}$ cannot be approved by t honest users and generate a valid (t, n) threshold ring signature TRS. Therefore, the adversary $\mathscr{A}$ needs to create an "honest looking" initial candidate transaction $\widetilde{tx_i^*} \neq tx_i^*$ such that $H(\widetilde{tx_i^*}) = H(tx_i^*)$, where H is a collision-resistant hash function. Next, the adversary $\mathscr{A}$ deceives the honest users, and the honest users ensure the initial candidate transaction $\widetilde{tx_i^*}$ by generating a threshold ring signature TRS. Then, the adversary $\mathscr{A}$ creates the final candidate transaction $\widetilde{tx_i^{**}} = \widetilde{tx_i^*} \| \mathrm{TRS}$ successfully and redacts the chain with the "honest looking" final candidate transaction $\widetilde{tx_i^{**}}$. However, the chance of creating such a transaction $\widetilde{tx_i^*}$ is negligible, since it violates the collision-resistance property of the hash function H. To sum up, the proof of Theorem 3 is established. □

5.4. Common Prefix. The property of common prefix is parameterized by $k \in \mathbb{N}$, which states that if there are two honest users, and they hold two different chains C_1 and C_2 respectively, the number of the far right different blocks of these two chains is at most k. The game is also executed between an adversary $\mathscr{A}$ and a simulator $\mathscr{B}$.

Definition 2. For any two different chains C_1 and C_2 held by two honest users U_1 and U_2 respectively, it holds that

$$\begin{aligned} C_2^k &\preccurlyeq C_1, \\ C_1^k &\preccurlyeq C_2. \end{aligned} \tag{3}$$

As shown in (3), C_1^k and C_2^k are denoted as the chain resulting from deleting the k far right blocks of chain C_1 and C_2.

However, Definition 2 cannot be used directly in our new redactable protocol Γ'. Consider that if a redaction request $r_{i,j}$ and the corresponding final candidate transaction tx_i^{**} were approved, an honest user U_1 updates the chain C_1 at time t_1 and he replaces the old transaction tx_i with tx_i^{**} in his local chain. However, another honest user U_2 may not update the chain C at time t_1 timely. In this case, $C_1^k \not\preccurlyeq C_2$ and $C_2^k \not\preccurlyeq C_1$, which violates Definition 3.

Therefore, a new definition was described as follows for the new privacy-preserving redactable protocol Γ'. Note that the redacted block is denoted as B_j^*.

Definition 3. For any two different chains C_1 and C_2 held by two honest users U_1 and U_2 respectively, it holds that

(1) $C_1^k \preccurlyeq C_2$ and $C_2 \preccurlyeq C_1$

(2) For any redacted block $B_j^* \in C_1^{(l_2 - l_1)+k}$ and $B_j^* \notin C_2^k$, or block $B_j^* \in C_2^{(l_2 - l_1)+k}$ and $B_j^* \notin C_1^k$, it satisfies $\Gamma'.\mathrm{validateCandTx}(\mathscr{C}, tx_i^{**}, j) = 1$

Theorem 4. *Γ' satisfies the property of k-editable common prefix if the public immutable blockchain satisfies the property of k-common prefix.*

Proof. Proof. Assume that the adversary $\mathcal{A}$ proposes a final candidate transaction tx_i^{**} to redact an honest block B_j in chain C_2. The chain C_2 is redacted by an honest user U_1 later. However, the adversary $\mathcal{A}$ cannot create another final candidate transaction $\widetilde{tx_i^{**}} \neq tx_i^{**}$ such that $H(\widetilde{tx_i^{**}}) = H(tx_i^{**})$ because of the collision-resistance property of hash function H. Since the final candidate transaction tx_i^{**} is accepted by the honest user U_1, it must be the case that tx_i^{**} contains a valid threshold ring signature TRS and is approved by most of the honest users in the system. To sum up, the proof of Theorem 4 is established. □

6. Experiments

Several experiments are executed to test the efficiency of our work. We give the time of block redaction and block generation and compare the efficiency of the proposed protocol with that of the previous ones. We mainly focus on the time-consuming operations in these protocols, and the most expensive operations for generating and redacting a block are separately solving the PoW puzzle and generating a threshold ring signature.

We conduct our experiments on a computer with a 64-bit Win 10 operating system, 8.0 GB RAM and Intel(R) Core(TM) i7-5500 CPU @ 2.4 GHz processor. We use the IntelliJ IDEA 2020 compiler and Java language to implement our programs. The JPBC 2.0.0 library is used to generate elliptic curve groups and pairings. The elliptic curve groups and pairings are used in the threshold ring signature scheme [22]. The AES algorithm is used to encrypt the transaction data.

We test multiple times to get the average values as the final measurement results from each experiment. It should be noted that the number of the consecutive zero of the hash prefixes is defined as the difficulty level. For example, the difficulty level is 5 when the number of consecutive zeros of the hash prefixes is 5.

6.1. Time Consumption of Generating a Block and Redacting a Transaction. We first test the computational overhead for generating a block. As shown in Figure 4, the overhead of generating a block is almost a constant, i.e., 2.894 s and 15.127 s, when the difficulty level is set as 4 and 6, respectively. The reason is that the overheads of generating a block depend on the difficulty of the POW puzzle instead of the percentage of the users participating in the redaction operation. We also test the overheads for redacting a transaction, which is much more important, and the time to redact a transaction actually determines the efficiency of our proposed scheme. As shown in Figure 5, the time consumption increases when the percentage of the users participating in a redaction operation grows, and the time consumption of redacting a transaction is not affected by the difficulty. The reason is that the time of redacting a transaction is mainly determined by the time of generating a (t, n) threshold ring signature, which depends on the value of threshold t.

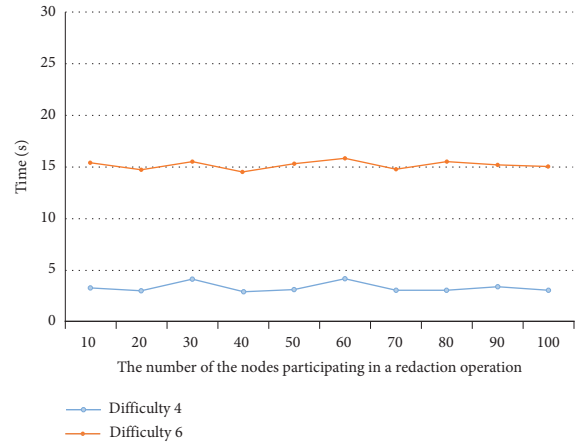

Figure 4: Time consumption of generating a block.

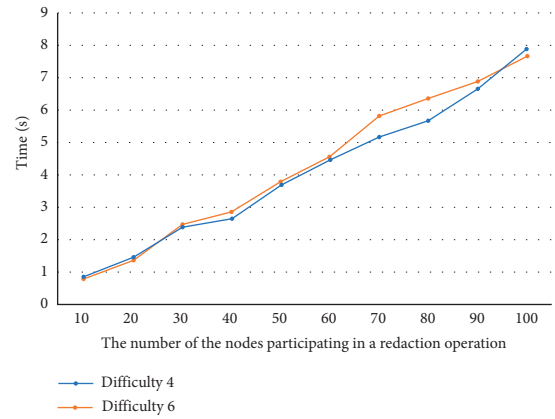

Figure 5: Time consumption of redacting a transaction.

In Table 1, we give the time of redacting a transaction when the difficulty level is 5 and the percentage of the users participating in the redaction operation is 80%. The experimental results indicate that the average time of generating a block and redacting a transaction is separately 4580 ms and 7088 ms, and the time of redacting a transaction is about twice that of generating a block. Therefore, redacting data on a block is efficient in the proposed protocol.

In order to describe the change of the block structure before and after the block is redacted more clearly, the block 23 is used as an example to illustrate what happens when a block is redacted. The original block information from block 22 to block 24 in the current blockchain is shown in Figure 6. Suppose that the second transaction in block 23 needs to be redacted. In order to redact the transaction, the owner of transaction tx_2 discloses the encrypted transaction data and the encryption key as the redacting reasons, and then a new transaction tx_2^* is generated to replace the old transaction tx_2. Moreover, the POW puzzle of the new block 23^* is solved. The users in the blockchain reached a consensus to

Table 1: The time consumption of block generation and redaction.

Time (ms)	Block1	Block2	Block3	Block4	Block5	Average
Block generation	4875	4350	4586	4109	4217	4580
Block redaction	7016	6889	6875	6959	6830	7088

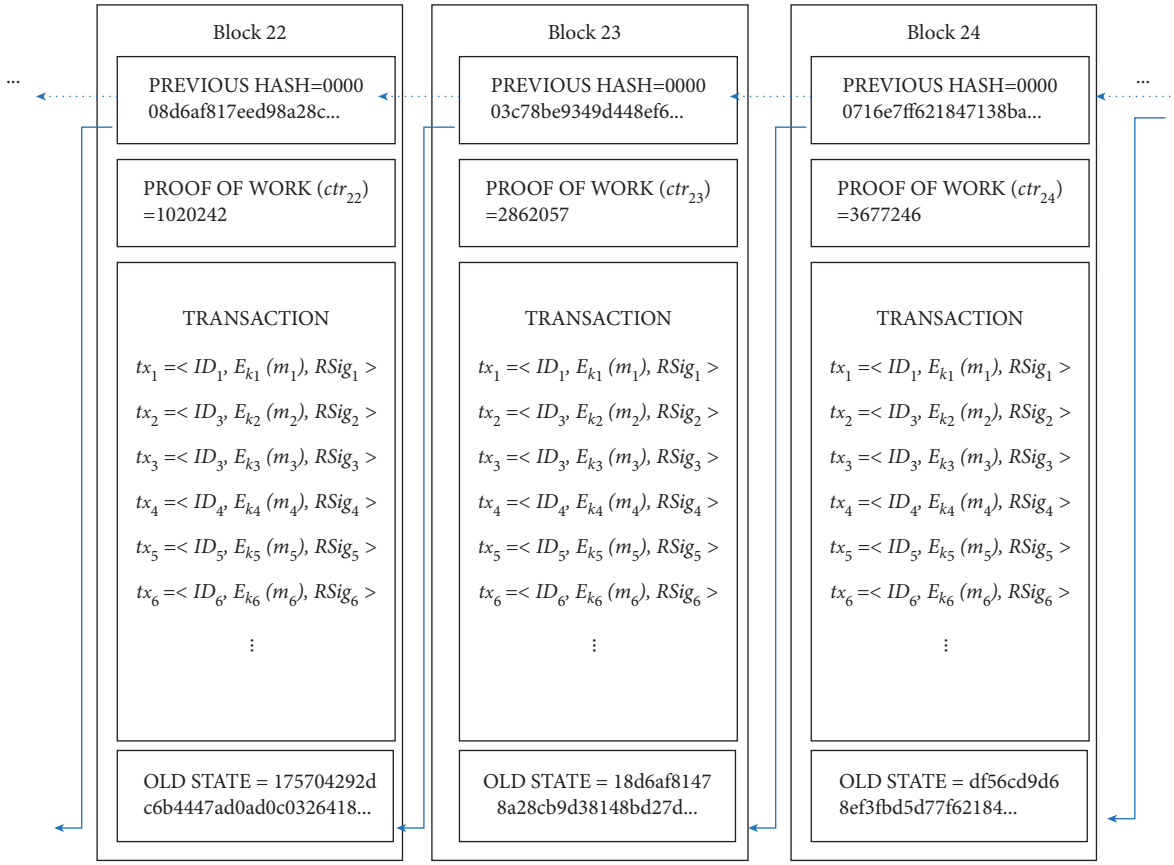

Figure 6: The original block information from block 22 to block 24.

redact the block and generate a threshold ring signature TRS to approve the redaction operation. The new redacted block information from block 22 to block 24 is shown in Figure 7.

6.2. Efficiency Comparison. This experiment is intended to compare the efficiency of our proposed redactable protocol with that of [11, 19], where a privacy-preserving deletable blockchain and a public redactable blockchain are proposed.

In the following experiments, we generate redactable and deletable chains. The length of these two chains and the number of block modifications are the same in each experiment. As shown in Figure 8, the number of block redactions or deletions ranges from 10 to 40, and the overhead ratio is from 76% to 84%, and thus the average time of a redaction in the proposed protocol is about 80 percent of that in the deletable chains [19]. The reason is that our protocol needs to generate a ring signature only once, and the protocol in [19] needs to generate a traceable ring signature twice. Therefore, a transaction redaction is more efficient in the proposed protocol than that of deleting a block in [19].

To compare the efficiency of our work and that of [11], we both generate privacy-preserving redactable and public redactable chains. The length of these two chains and the number of block modifications are kept the same. As shown in Figure 9, the number of redactions ranges from 10 to 40, and the overhead ratio is from 30% to 41%, and thus the average time of redacting a transaction in the proposed protocol is about three times that in the public chains [11]. The reason is that our protocol uses more time-consuming operations, including the threshold ring signature and symmetric encryption in the redacting process to protect identities and data of the users. Therefore, the proposed protocol spends more time than the protocol in [11].

Next, we give a time comparison of redaction for these three protocols, and the results are shown in Table 2. We can see that, in the protocol of [19], it takes an average of 8904 ms to delete a block. However, the protocol can only delete the whole block by the data owner, and it cannot redact a single

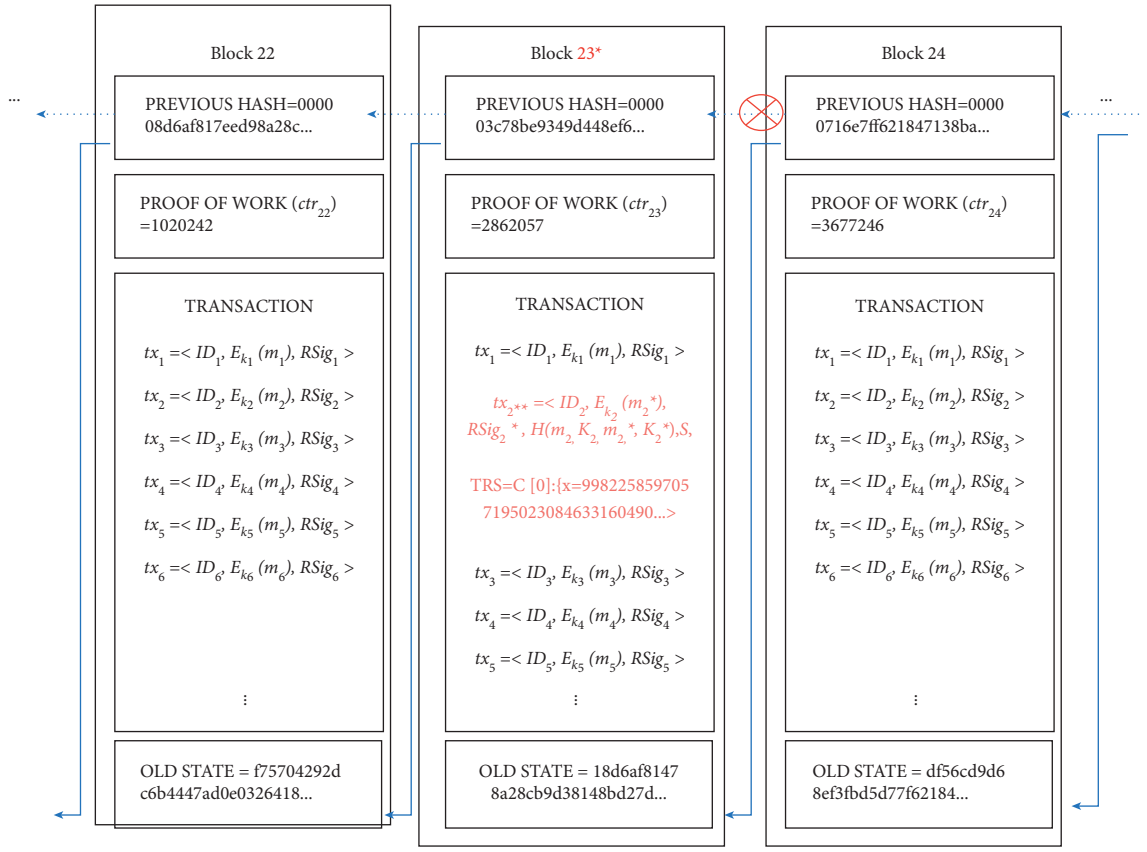

Figure 7: The new redacted block information from block 22 to block 24.

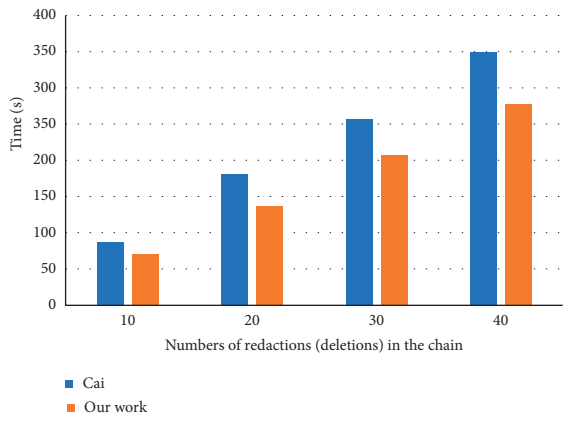

Figure 8: Block redaction overhead compared to deletable blockchain [19].

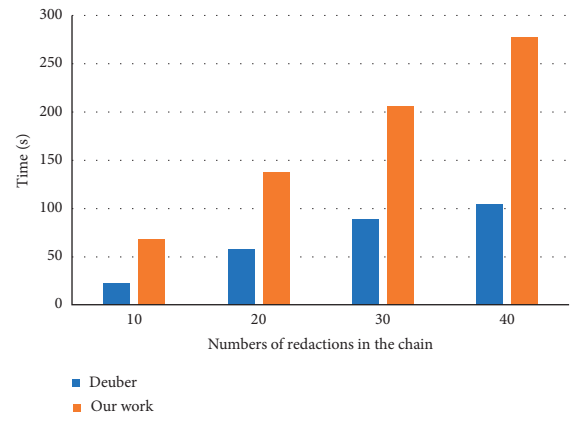

Figure 9: Transaction redaction overhead compared to the public redactable blockchain [11].

transaction in the block. In our protocol, it takes an average of 7088 ms to redact a transaction, which is about 79.6% of the time for deleting a block in [19]. In the protocol of [11], it takes an average of 2552 ms to redact a transaction, which is about 36.0% of the time for redacting a transaction in the proposed redactable blockchain. However, the identities of the users and the transaction data are all public in the protocol of [11]. Thus, the proposed protocol can realize the redaction on the transaction-level, while the identities of the users and transaction data are also private.

Finally, we compare the proposed protocol with those in [11, 19], and the results are shown in Table 3. It is concluded that our protocol provides the function of transaction redaction, transaction deletion without disclosing the identities of the users and the transaction data. Although the protocol in [19] realizes the deletion of block data, the transaction redaction is not allowed. In the protocol of [11], transaction redaction is allowed, but the identities of the users and transaction data are public. The proposed protocol constructs a transaction-level redactable blockchain, and the

Table 2: The time comparison for block redaction.

Time (ms)	Block 1	Block 2	Block 3	Block 4	Block 5	Average
Cai et al. [19]	8739	9069	8630	8754	9328	8904
Deuber et al. [11]	2194	2889	2970	2645	2072	2552
Our work	7016	6889	6875	6959	6830	7088

Table 3: The comparison of three redactable blockchain protocols.

	Redaction	Deletion	Identity privacy	Data privacy	Transaction level	Block level
Cai et al. [19]	×	✓	✓	✓	×	✓
Deuber et al. [11]	✓	✓	×	×	×	✓
Our work	✓	✓	✓	✓	✓	×

users only need to replace a single transaction to complete the data redaction instead of replacing the entire block. However, the protocol [11, 19] can only achieve block-level redactable blockchain.

7. Conclusion

In this paper, we propose a privacy-preserving redactable blockchain based on the old state of the block. A symmetric encryption algorithm and a threshold ring signature are separately used to protect the transaction data and the users' identities during the process of redaction. All of the users can check whether a redaction operation is valid or not according to the threshold ring signature and the old link of the redacted block. The experiments' results indicate that the proposed protocol is efficient and effective, and the identities of the users and the transaction data are also private.

Data Availability

All data are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. U1736120) and Natural Science Foundation of Shanghai (20ZR1419700).

References

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," manubot, Technology Report, 2019.

[2] T. Jiang, H. Fang, and H. Wang, "Blockchain-based internet of vehicles: distributed network architecture and performance analysis," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4640–4649, 2018.

[3] H. N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: a survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[4] E. F. Jesus, V. R. L. Chicarino, C. V. N. D. Albuquerque, and A. A. D. A. Rocha, "A survey of how to use blockchain to secure internet of things and the stalker attack," *Security and Communication Networks*, vol. 2018, Article ID 9675050, 27 pages, 2018.

[5] R. Kamath, "Food traceability on blockchain: walmart's pork and mango pilots with ibm," *The Journal of the British Blockchain Association*, vol. 1, no. 1, 2018.

[6] 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, Article ID 8315614, 15 pages, 2019.

[7] T. McGhin, K. K. R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: research challenges and opportunities," *Journal of Network and Computer Applications*, vol. 135, pp. 62–75, 2019.

[8] J. Wang, L. Wu, H. Wang, K. K. R. Choo, and D. He, "An efficient and privacy-preserving outsourced support vector machine training for internet of medical things," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 458–473, 2020.

[9] S. Zeng, Y. Yuan, X. C. Ni, and F. Y. Wang, "Scaling blockchain towards bitcoin: key technologies, constraints and related issues," *Acta Automatica Sinica*, vol. 45, no. 6, pp. 1015–1030, 2019.

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

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

[12] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, pp. 45–58, 2019.

[13] S.-F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen, "Ringct 2.0: a compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero," in *Proceedings of the European Symposium on Research in Computer Security*, pp. 456–474, Springer, Cham, Oslo, Norway, August 2017.

[14] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: anonymity for bitcoin with accountable mixes," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 486–504, Springer, Berlin, Heidelberg, November 2014.

[15] E. B. Sasson, A. Chiesa, C. Garman et al., "Zerocash: decentralized anonymous payments from bitcoin," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 459–474, IEEE, Berkeley, CA, USA, May 2014.

[16] T. H. Yuen, S. Sun, J. K. Liu et al., "Ringct 3.0 for blockchain confidential transaction: shorter size and stronger security," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 464–483, Springer, St. Kitts and Nevis, February 2019.

[17] G. Ateniese and B. D. Medeiros, "Identity-based chameleon hash and applications," in *Proceedings of the International Conference on Financial Cryptography*, pp. 164–180, Springer, FL, USA, February 2004.

[18] D. Derler, K. Samelin, D. Slamanig, and C. Striecks, "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 2019*, CA, USA, April 2019.

[19] X. Cai, Y. Ren, and X. Zhang, "Privacy-protected deletable blockchain," *IEEE Access*, vol. 8, pp. 6060–6070, 2020.

[20] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: analysis and applications," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 281–310, Springer, Sofia, Bulgaria, April 2015.

[21] S. S. Chow, L. C. Hui, and S.-M. Yiu, "Identity based threshold ring signature," in *Proceedings of the International Conference on Information Security and Cryptology*, pp. 218–232, Springer, Shanghai, China, November 2004.

[22] T. H. Yuen, J. K. Liu, M. H. Au, W. Susilo, and J. Zhou, "Threshold ring signature without random oracles," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pp. 261–267, Hong Kong, China, March 2011.