Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1040662, 11 pages
https://doi.org/10.1155/2022/1040662

Research Article

Blockchain Data Privacy Protection and Sharing Scheme Based on Zero-Knowledge Proof

Tao Feng[1], **Pu Yang**[1], **Chunyan Liu**[2], **Junli Fang**[1] and **Rong Ma**[1]

[1]*School of Computer and Communication, Lanzhou of University of Technology, Lanzhou 730050, China*
[2]*School of Economics and Management, Lanzhou of University of Technology, Lanzhou 730050, China*

Correspondence should be addressed to Pu Yang; ypu97717@163.com

Received 3 November 2021; Accepted 21 January 2022; Published 23 February 2022

Academic Editor: Jinguang Han

The data generated in the Industrial Internet of Things (IIoT) has important research value. In the process of data sharing, data privacy, security, and data availability are important issues that cannot be ignored. This paper proposes a blockchain privacy protection scheme based on zero-knowledge proof to realize the secure sharing of data among data owners, cloud service providers, and semitrusted cloud servers. First, the method of combining zero-knowledge proof and smart contract is used to verify the availability of data between the data owner and the cloud service provider under the premise of protecting data privacy. Second, proxy reencryption technology is used to realize the secure sharing of data among authorized cloud service providers. In addition, data sharing transaction information between multiple parties and data hashes with digital signatures are stored on the blockchain to achieve public and verifiable data sharing information and data validity. Finally, the theoretical analysis of the scheme shows that the scheme meets the confidentiality requirements of security, integrity, and validity.

1. Introduction

Since the 21st century, the Internet has given traditional industries the explosive growth of data in the Industrial Internet of Things [1, 2]. The massive amount of data generated in different fields (such as smart home, smart city, and smart manufacturing) has extremely high research value, which has aroused research interest in industry and academia. How to share data safely and efficiently, use data to provide users with better and convenient services, and improve user experience has become a widespread concern today. However, most of the data generated by IIoT is the user's private data. In the process of data sharing, it is necessary to ensure the privacy, integrity, and validity of the data [3–5]. For example, sensitive and private data is tampered with or leaked during the sharing process. Data owners may provide irrelevant or false data to cloud service providers. Cloud service providers do not want data owners to provide information to other research institutions. Therefore, the following problems still exist in the data sharing of the Industrial Internet of Things. (1) There is lack of protection of data privacy and security in the data sharing process. (2) The data recipient cannot ensure that the data obtained is valid and relevant information. (3) Data integrity and data transaction records cannot be verified and traced during the data sharing process. Therefore, due to the above-mentioned problems, there is an urgent need for a solution to realize data sharing while protecting privacy and security.

Zero-knowledge proof is a cryptographic technology, which can make the verifier believe that a certain assertion is correct without providing any valuable information to the verifier. Zero-knowledge succinct noninteractive knowledge argumentation (zk-SNARKs) is one of the tools for generating zero-knowledge proofs. In the blockchain transaction platform, it is used in cryptocurrencies such as Zcash [6] and ZETH [7] to hide private information such as the address of the sender and receiver of the transaction and the transaction amount. In the data sharing between the cloud service provider and the data owner, zero-knowledge proof combined with smart contract technology can realize data availability verification between the two parties' data

transactions and ensure the provision of effective data information.

Blockchain is an effective method to solve verifiable and traceable transactions due to its decentralization, immutability, traceability, and executable smart contracts. Due to the characteristics of its distributed data ledger, it is widely used in multiple scenarios such as virtual currency, electronic bidding, and Industrial Internet of Things. In terms of addressing data privacy, blockchain can be combined with a variety of cryptographic methods, for example, attribute encryption [8], homomorphic encryption [9], searchable encryption, and proxy reencryption combined [10], to achieve the protection of data privacy and identity privacy on the blockchain.

In the data sharing scheme based on blockchain, some researchers have implemented data sharing schemes for individual users. However, these solutions focus on the aggregation of data and the balance between data privacy and data accessibility in the process of data sharing transactions, and data transmission between multiple entities cannot ensure user data privacy in the entire process. In response to these existing problems, this paper proposes a blockchain data privacy protection and sharing scheme based on zero-knowledge proof. It solves the problems of data privacy security, data availability and consistency, and data transaction traceability in data sharing.

The main research contributions of this paper are as follows.

(1) In multientity data sharing, a zero-knowledge proof-based blockchain data privacy protection and sharing scheme is proposed to achieve privacy protection. Use proxy reencryption technology to ensure data sharing between cloud service providers and data owners. Realize data sharing, traceability, and verifiability among multiple entities based on blockchain characteristics

(2) A method of combining zero-knowledge proof and smart contract is proposed. The data owner can prove that the data meets the requirements of the cloud service organization without revealing any data privacy, realize the consistency and availability of the data in the sharing process, and protect the interests of both parties. After the verification is passed, the improved consensus algorithm enables the nodes to reach consensus directly and faster

(3) Through security analysis and comparison with other solutions, this solution realizes the sharing of data among multiple entities under the premise of not revealing any data privacy, and the consistency, availability, and traceability, and verifiable characteristics of the sharing process during the sharing process. And it has better consensus efficiency

2. Related Work

In a data sharing scheme based on cloud services, it relies on some encryption methods to protect data privacy. However, the data is difficult to trace and verify, and the data is easy to be stolen and tampered. Blockchain can be used to solve some of the current problems in data sharing due to its decentralization, immutability, traceability, and other characteristics. In the data sharing scheme based on blockchain [11, 12], data privacy protection combined with data encryption mainly uses encryption methods such as attribute encryption and proxy reencryption.

In the research of data sharing based on cloud services, Muthusenthil et al. [13] proposed a new secure data sharing reencryption scheme based on trusted institutions, using proxy reencryption methods to ensure data privacy and security, with better performance. However, the solution cannot guarantee user identity privacy and does not have the traceability of transactions and data. Mahakalkar and Sahare [14] proposed SAPA, a privacy protection authentication protocol based on sharing authority, which uses reencryption to realize data sharing between multiple users. The use of an access request matching mechanism realizes the user's identity is private, but cannot guarantee the traceability of data and transactions. Wang et al. [15] proposed an identity-based data sharing audit scheme, which uses an information-hiding mechanism and a security mechanism that simplifies the signature algorithm to protect sensitive information and prevent malicious managers. However, the validity and consistency of the data cannot be guaranteed. Cheng et al. [16] proposed a reliable and efficient data sharing solution for the Industrial Internet of Things (IIoT). The scheme is based on an adaptive decentralized inadvertent transmission protocol, combined with zero-knowledge proof technology, so that the private key of the data recipient can be hidden from the data owner during the data sharing process. The traceability of data is realized, but the traceability of transactions cannot be realized.

In the research of blockchain-based data sharing solutions, Chowdhury et al. [17] proposed a notarization service framework based on blockchain-based personal data storage and sharing. This framework will ensure the authenticity of real-time shared data, and the transaction privacy is provided in the chain network. However, the complete traceability of the data is guaranteed in the process, but the privacy of the data cannot be guaranteed. Lu et al. [18] designed and implemented a blockchain-authorized secure data sharing architecture, combined with federal learning combined with privacy protection, transformed data sharing problems into machine learning problems, and maintained data privacy. However, the traceability of the transaction and the integrity of the data cannot be guaranteed. Wang et al. [19] proposed a blockchain-based security and privacy protection electronic medical record sharing protocol, which combines searchable encryption and conditional proxy reencryption to achieve data security, privacy protection, and access control. However, the validity of the data cannot be guaranteed. Sani et al. [20] proposed a high-performance, scalable blockchain that enhances the security and privacy of IIoT, using time-based zero-knowledge proof and authentication encryption to perform mutual authentication between multiple attributes. The evaluation from the three aspects of security, privacy, and performance shows that the scheme is safe, and the computational complexity and delay performance are significantly reduced. The privacy of

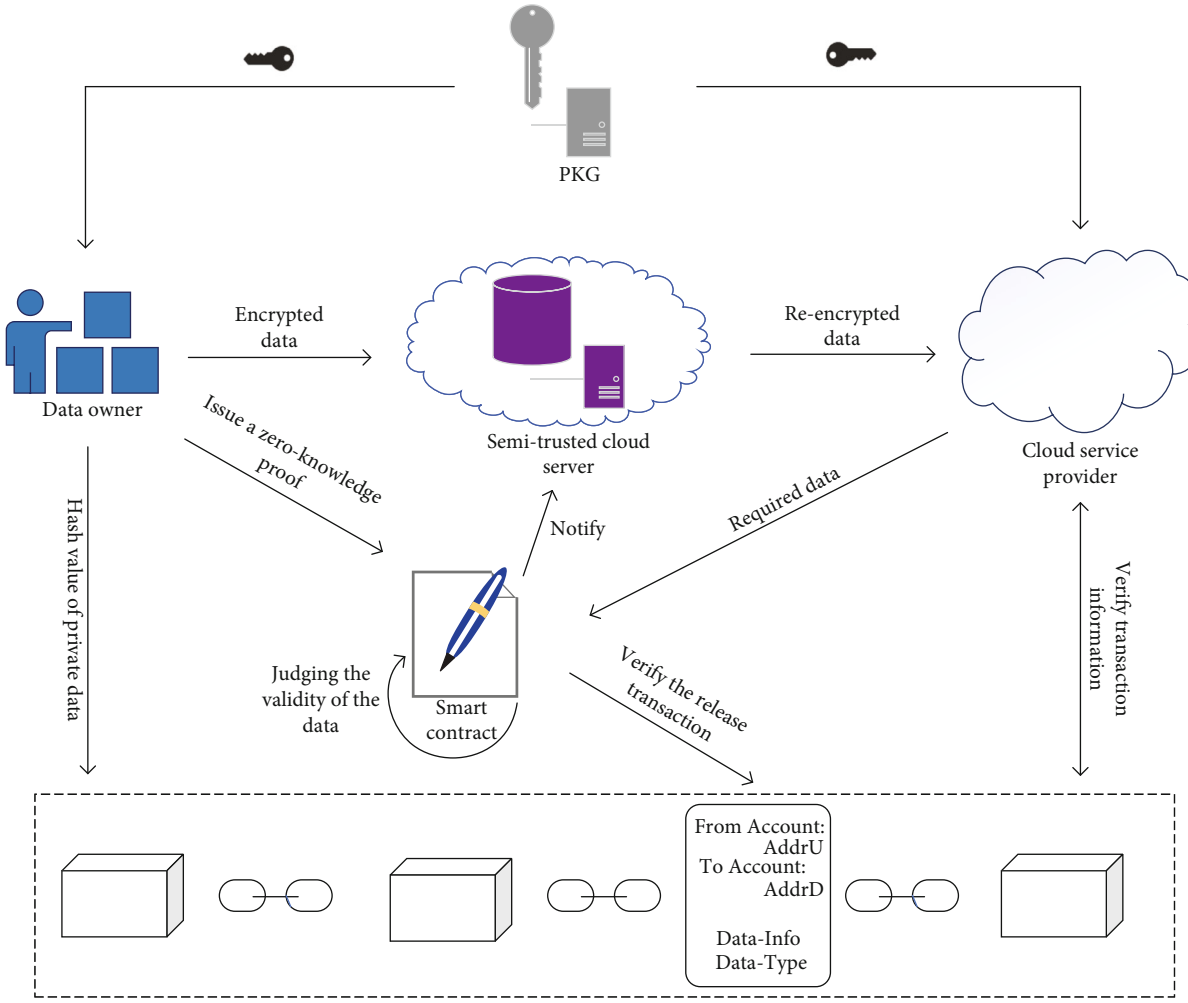

Figure 1: Blockchain data privacy protection and sharing scheme model based on zero-knowledge proof.

identity and data is guaranteed, but the traceability of transactions cannot be achieved. Shen et al. [21] proposed a reliable sharing and collaboration model based on blockchain. Data owners, miners, and third parties share data through blockchain and record through smart contracts. Participants can use private clouds or public clouds to obtain and store data sharing. The identity privacy of data participants is guaranteed, but the content privacy of data cannot be guaranteed. Kouicem et al. [22] proposed a decentralized and anonymous vehicle data sharing scheme, allowing each vehicle to anonymously verify each data record without revealing the identity of the vehicle sharing this data. Each vehicle sends a certificate to the data record, which uses zero-knowledge proof (ZKP) to anonymously combine the data record and the user's identity. Identity privacy is guaranteed, but the traceability of data transactions cannot be achieved. Manzoor et al. [23] proposed a blockchain-based IoT data sharing scheme. Use proxy reencryption to store and share Internet of Things data in a cloud proxy server, and establish smart contracts between sensors and data consumers without the involvement of a trusted third party. The privacy of the data is guaranteed, but the validity and consistency of the data cannot be guaranteed.

From the above scheme, we can see that the blockchain-based cloud data sharing scheme has achieved certain research results, and a variety of data sharing schemes have been proposed using blockchain technology and cryptographic methods. However, the consistency and availability of data in data sharing and the traceability and verifiability of data sharing transactions between multiple entities have not been effectively improved.

3. Problem Description

The solution proposed in this article combines blockchain, agent heavy intelligence, smart contracts, and zk-SNARK technology to achieve privacy protection and data security sharing among data owners, cloud service organizations, and semitrusted cloud servers. The system model of our proposal is shown in Figure 1.

It includes 6 participating entities: (1) data owner, (2) cloud service organization (CSP), (3) semitrusted cloud server (semitrusted CS), (4) private key generator (PKG), (5) smart contract, and (6) blockchain (Blockchain). Their functions are described as follows.

(i) Data owner: data owners have the right to securely own and conditionally share their information and data and can obtain corresponding benefits as remuneration during the sharing process

(ii) Cloud service organization (CSP): as a consumer of private data, cloud service organization needs to collect and analyze private data. They issue corresponding privacy data requirements by entrusting smart contracts. But at the same time, they do not believe that the data provided by the data owner meets their needs, so they use smart contracts to ensure the consistency and effectiveness of the data requirements

(iii) Semitrusted cloud server: as a semitrusted entity, it needs to store the original ciphertext of the data owner and is responsible for converting it into intermediate ciphertext, which will be handed over to the cloud service organization after verification and decrypted by its private key

(iv) Private key generator (PKG): it is a completely trusted entity that needs to generate master keys and system parameters and distributes public keys and keys to data owners and cloud service organizations

(v) Smart contract: smart contracts are responsible for predeclaring the requirements and the specific structure of private data and guaranteeing certain data benefits. Automatically judge the validity of the zero-knowledge proof without the participation of a third party

(vi) Blockchain: responsible for reaching a consensus on data transactions. Store the hash of private data in the blockchain to ensure the immutability and traceability of the data, which is the evidence for data disputes

4. Security Model

4.1. Definition. Based on the DBDH assumption, under the random oracle model, if there is a negligible function $\varepsilon(\kappa)$ for any polynomial time adversary A, such that $\mathrm{Adv}^{\mathrm{CPA}}_{\prod A}(\kappa) \leq \varepsilon(\kappa)$, then the encryption algorithm $\prod$ is indistinguishable under the selected plaintext attack, which is called IND-CPA (indistinguishability-chosen plaintext attack) security.

4.2. Initialization. Challenger B runs the initialization algorithm to generate public parameters and master key (PK , MSK) and sends the public parameters PK to the adversary A.

Stage 1: the adversary A sends a key generation request to the challenger, and the challenger B generates a key pair $(\mathrm{PK}_u, \mathrm{SK}_u)$ and sends it to A.

Challenge phase: the adversary A sends two messages of the same length, m_0 and m_1, to the challenger B. The challenger B randomly selects $\sigma \in (0, 1)$ and sends $\mathrm{Enc} = E_n(\mathrm{PK}, \mathrm{PK}_u, m_\sigma)$ to the adversary.

Stage 2: the adversary A repeats the request phase 1.

The adversary A outputs a guess value of $\sigma' \in (0, 1)$; if $\sigma = \sigma'$, the adversary A wins the game. The advantage of the adversary A can be defined as $\mathrm{Adv}^{\mathrm{CPA}}_{\prod A}(\kappa) = |\Pr[\sigma' = \sigma] - (1/2)|$.

5. Blockchain Data Privacy Protection Scheme Based on ZKP

5.1. Scheme Steps. As shown in Figure 2, after each entity is registered in the blockchain, the private key generation center assigns a common private key pair to the user. The cloud service provider generates a zero-knowledge proof π' of the required data through zk-SNARK, sends the calculation results R' and hash value h' to the smart contract, and records and publishes the required keywords in the blockchain. The data owner generates an encrypted ciphertext according to the needs of the cloud service provider, sends it to the semitrusted cloud server, and records the signed hash in the blockchain. At the same time, the zero-knowledge proof π generated by the private data, the calculation result R, and the hash value h are sent to the smart contract for automatic comparison. After passing the zero-knowledge proof verification, the data owner is notified to use the public key PK_d of the cloud service organization to execute the reencryption algorithm to generate the reencryption key $\mathrm{PK}_{u \longrightarrow d}$ and send it to the semitrusted proxy cloud server through the public key of the cloud service provider. The semitrusted proxy cloud server executes the reencryption algorithm to convert the ciphertext C_{PK_u} into the intermediate ciphertext $C_{\mathrm{PK}_{u \longrightarrow d}}$ and then sends the intermediate ciphertext to the cloud service provider. The cloud service provider uses the private key SK_d to execute the decryption algorithm to obtain the required private data for verification based on the information on the blockchain. After the smart contract is passed, the data sharing information transaction is submitted to the verification node, and the RBFT consensus algorithm is used to verify it and then publish it on the blockchain. The symbols used in this article are shown in Table 1.

6. Specific Structure

The specific construction process of the scheme is divided into the following ten stages.

6.1. Join the Network Phase. Equations should be provided in a text format, rather than as an image. Microsoft Word's equation tool is acceptable. Equations should be numbered consecutively, in round brackets, on the right-hand side of the page. They should be referred to as Equation 1, etc. in the main text.

6.2. Data Initialization Phase. First, PKG chooses to input a security parameter λ, selects prime number to generate multiplication cycles G_1 and G_1, and selects four hash function groups $H_1 : \{0,1\}^* \longrightarrow \{0,1\}^k$, $H_2 : \{0,1\}^* \longrightarrow G_1$, $H_3 : G_2 \longrightarrow \{0,1\}^k$, and $H_4 : \{0,1\}^k \times \{0,1\}^l \longrightarrow \mathbb{Z}_p^*$.

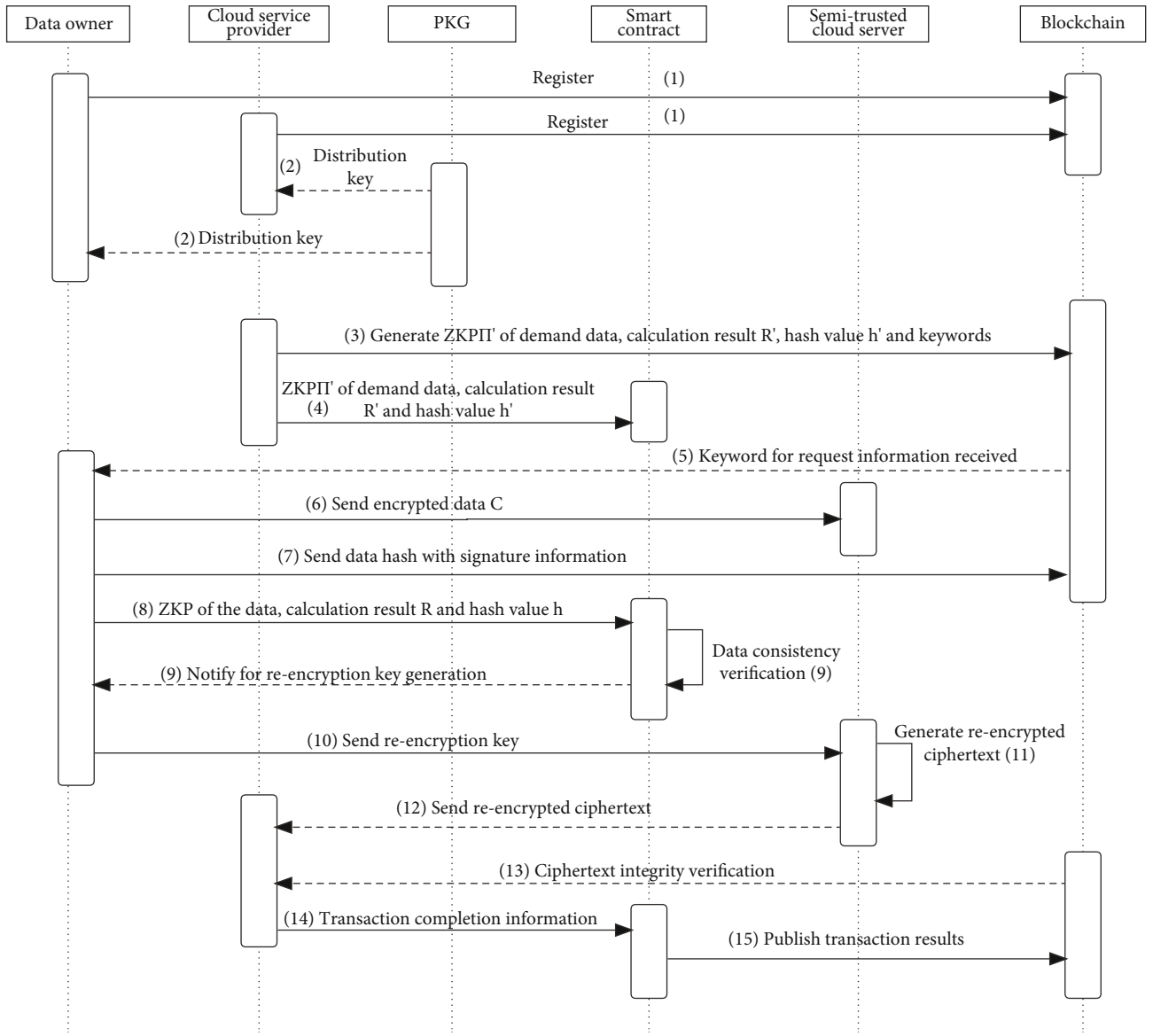

Figure 2: Timing diagram of blockchain data privacy protection and sharing scheme based on ZKP.

Table 1: Notations.

Symbols	Definitions
λ	Security parameter
PKG	Private key generation center
D	Data owner's private data
PK_u, PK_d	The public key of the data owner and cloud service provider
SK_u, SK_d	The private key of the data owner and cloud service provider
$RK_{u \longrightarrow d}$	Reencryption key
$C_{PK_{u \longrightarrow d}}$	Reencrypted ciphertext
σ_a	Digital signature
π	Zero-knowledge proof
EK_C	Generate the key for zero-knowledge proof
VK_C	Key to verify zero-knowledge proof

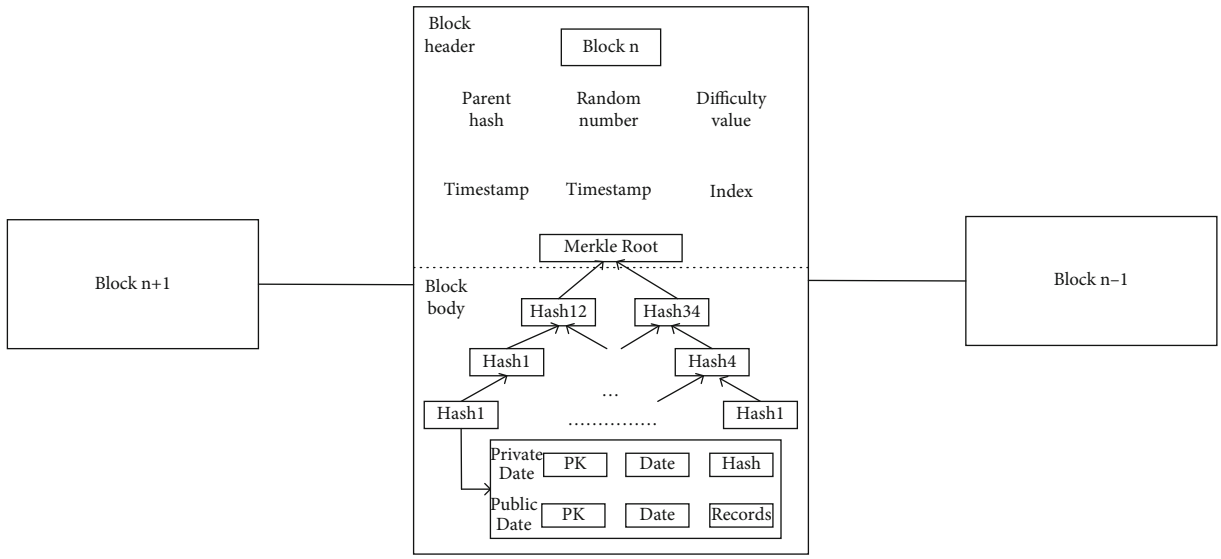

Figure 3: Transaction form.

Define the bilinear mapping $e: G_1 \times G_2 \longrightarrow G_2$. Then, PKG randomly selects $a, b, c \in \mathbb{Z}_p^*$, and $g, h \in G_1$ are two different generators of G_1. Generate public parameters and master key $\text{Setup}(1^\lambda) \longrightarrow (\text{PK}, \text{MSK})$, where $\text{PK} = (p, G_1, G_2, e, g, h, H_1, H_2, H_3 H_4)$, $\text{MSK} = (a, b, c)$.

PKG uses its identity ID_u provided by the data owner to generate its public and private key pair $\text{KeyGen}(\text{MSK}, \text{PK}, \text{ID}_u) \longrightarrow (\text{PK}_u, \text{SK}_u)$, and the cloud service provider obtains the key pair in the same way. PKG randomly selects parameters $t, x, y, z \in \mathbb{Z}_p^*$ to calculate the private key of the data owner as follows:

$$\begin{aligned} A_1 &= \frac{c+t}{a+b\cdot \text{ID}_u} \; A_2 = h^t \; A_3 = g^t, \\ B_1 &= \frac{a+x}{a+b\cdot \text{ID}_u} \; B_2 = \frac{b+x}{a+b\cdot \text{ID}_u} \; B_3 = \frac{z}{a+b\cdot \text{ID}_u}, \\ D_1 &= h^x \; D_2 = h^y \; D_3 = h^z. \end{aligned} \tag{1}$$

Among them, (A_1, A_2, A_3) is used to recover the ciphertext, and $(B_1, B_2, B_3, D_1, D_2, D_3)$ is used to generate the reencryption key.

6.3. Smart Contract Release Phase. The cloud service provider uses zk-SNARKs to generate a zero-knowledge proof π' that includes some of its attribute requirements, the calculation result R', and the hash value is recorded in the smart contract and at the same time publish some keywords for data requirements. The generation process of the zero-knowledge proof will be described below from the perspective of the data owner, and the zero-knowledge proof process of the cloud service organization is similar.

6.4. Encryption Phase. After the data owner generates the private data, it will encrypt private data $D = \langle d_1, d_2, \cdots, d_n \rangle$ by $\text{Encrypt}(\text{PK}, \text{PK}_u, \langle d_1, d_2, \cdots, d_n \rangle) \longrightarrow C_{\text{PK}_u}$, where $C_{\text{PK}_u} = (c_{pk_1}, c_{pk_2}, \cdots, c_{pk_n})$ is that PKG randomly selects $r, s \in \mathbb{Z}_p^*$ to calculate the following parameters.

$$\begin{aligned} c_{pk_1} &= D \cdot e(g,h)^{c(r+s)}, \\ c_{pk_2} &= g^r, \\ c_{pk_3} &= h^s, \\ c_{pk_1} &= e(g,h)^{(a+b\cdot ID_u)(r+s)}. \end{aligned} \tag{2}$$

Then, the data owner uploads the ciphertext C_{PK_u} to the semitrusted proxy cloud server for storage. We assume that the semitrusted proxy cloud server will not modify the data of the data owner without authorization, and it will perform the operations we set honestly.

6.5. Data Record On-Chain Phase. The data owner will store the hash value and digital signature of the data record on the blockchain platform, and the private data will be encrypted and stored on the proxy cloud server. The data owner will submit the hash value of his private data $D = \langle d_1, d_2, \cdots, d_n \rangle$ to generate the transaction form shown in Figure 3 and attach his digital signature $\sigma_a = \text{Authsign}(\text{SK}_u, H(\langle d_1, d_2, \cdots, d_n \rangle))$ to it. When the transaction is verified by the verification node, it is recorded in the blockchain.

6.6. Generate Zero-Knowledge Proof Phase. When the data owner's private data meets the keyword requirements provided by the cloud service provider, the data owner attaches his digital signature and local time information to the private data and submits it to zk-SNARKs to generate a zero-knowledge proof. The construction process is as follows.

Step 1. According to the private data $D = \langle d_1, d_2, \cdots, d_n \rangle$ of the data owner ID_u, the local time T generates auxiliary information $\delta = (D, T, \text{ID}_u)$.

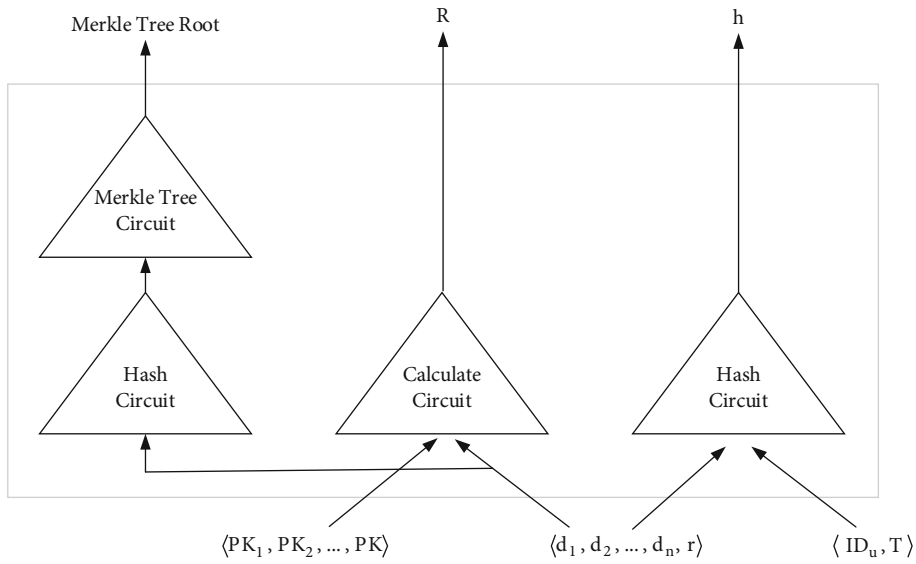

Figure 4: Circuit structure diagram.

Step 2. Select the random number r and the auxiliary information $\delta = (D, T, \mathrm{ID}_u)$ to calculate the hash value $H(\delta, r)$ and then generate the digital signature $\sigma_a = \mathrm{Authsign}(\mathrm{SK}_p, H(\delta, r))$ with the private key of the data owner.

Step 3. The data owner constructs the circuit $C : \mathbb{F}^n \times \mathbb{F}^h \longrightarrow \mathbb{F}^l$. Circuit input public parameters $\langle \mathrm{PK}_1, \mathrm{PK}_2, \cdots, \mathrm{PK}_n \rangle$, private data $D = \langle d_1, d_2, \cdots, d_n, r \rangle$, data owner identification information $\langle \mathrm{ID}_u, T \rangle$, where T and r are timestamps and random numbers, respectively. The output result R and the hash value h verify the authenticity and availability of the data.

$$C(\langle d_1, d_2, \cdots, d_n, r \rangle) \longrightarrow (R, h). \tag{3}$$

The circuit structure used in this paper is given in Figure 4.

Step 4. Enter the security parameter λ and the circuit C in the calculation task to calculate the key pair $(\mathrm{EK}_C, \mathrm{VK}_C)$, where EK_C is used to generate the zero-knowledge proof and VK_C is used to verify the zero-knowledge proof.

$$\mathrm{ZKPKeyGen}\left(1^{\lambda}, C\right) \longrightarrow (\mathrm{EK}_C, \mathrm{VK}_C). \tag{4}$$

Step 5. Prove algorithm consists of the generated key EK_C of the zero-knowledge proof, the private data D of the data owner, and the calculation result (R, h) in Step 3 together to generate the zero-knowledge proof π.

$$\mathrm{Prove}(\mathrm{EK}_C, D, R, h, \sigma_a) \longrightarrow \pi. \tag{5}$$

6.7. Zero-Knowledge Proof Verification Phase. The data owner submits the zero-knowledge proof to the smart contract. Then, the smart contract will automatically verify whether the zero-knowledge proof meets the requirements of the cloud service provider.

$$\mathrm{Verify}(\mathrm{VK}_C, \mathrm{PK}_u, \pi, R, h, \sigma_a) \longrightarrow (0, 1). \tag{6}$$

The smart contract first uses the public key of the data owner to verify its signature and then uses the verification key of zk-SNARKs to verify the zero-knowledge proof. After the verification is passed, the smart contract will automatically compare the zero-knowledge proof π of the data owner, the calculation result R, the hash value h, the zero-knowledge proof π' of the cloud service organization, the calculation result R', and the hash value h'. After the verification is completed, if the verification is correct, output 1; otherwise, output 0.

6.8. Reencryption Phase. After the verification is passed, the data owner uses the public key provided by the cloud service organization to generate the conversion key.

$$\mathrm{ReKeyGen}(\mathrm{PK}, \mathrm{SK}_u, \mathrm{PK}_d) \longrightarrow \mathrm{RK}_{u \longrightarrow d}. \tag{7}$$

PKG randomly selects the parameter $k_1, k_2 \in \mathbb{Z}_p^*$ and calculates as follows:

$$\begin{aligned} rk_1 &= (k_1 B_3 + B_1) + (k_2 B_3 + B_2) * \mathrm{ID}_u, \\ rk_2 &= \left(D_1 D_3^{k_1}\right)\left(D_2 D_3^{k_2}\right)^{\mathrm{ID}_u}, \\ \mathrm{RK}_{u \longrightarrow d} &= (rk_1, rk_2). \end{aligned} \tag{8}$$

The semitrusted proxy cloud server will convert the ciphertext into an intermediate ciphertext that can be decrypted by the cloud service organization after receiving the conversion key encrypted by the public key of the data owner. The proxy cloud server sends the intermediate

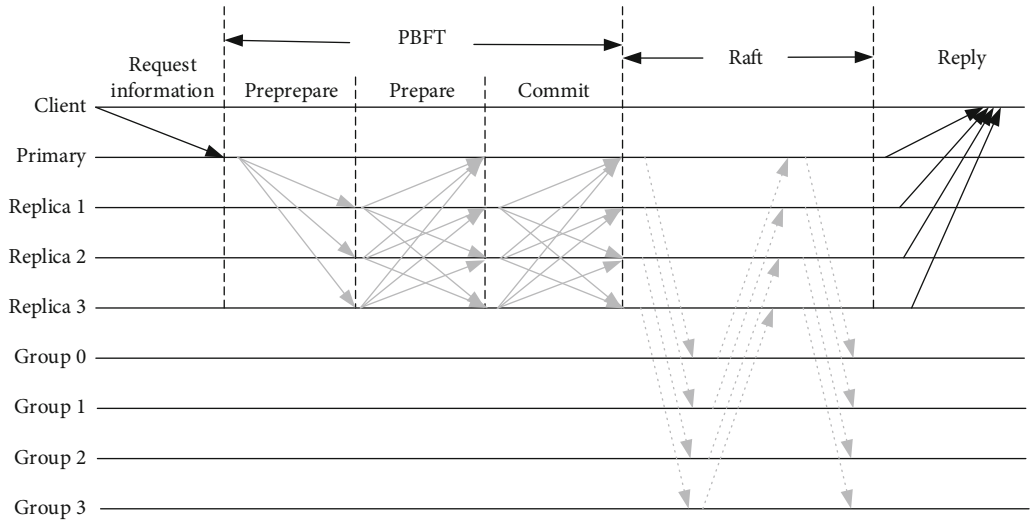

Figure 5: RBFT consensus process.

ciphertext $C_{\mathrm{PK}_{u\longrightarrow d}}$ to the cloud service organization.

$$\begin{aligned} c'_{pk_1} &= c_{pk_1}, \\ c'_{pk_2} &= c_{pk_2}, \\ c'_{pk_3} &= \frac{c_{pk_3}^{k_1}}{e\left(c_{pk_2}, rk_2\right)}, \\ C_{\mathrm{PK}_{u\longrightarrow d}} &= \left(c'_{pk_1}, c'_{pk_2}, c'_{pk_3}\right). \end{aligned} \tag{9}$$

6.9. Decryption Phase. When the cloud service provider receives the intermediate ciphertext obtained from the proxy cloud server, it decrypts the ciphertext with its private key. During the data sharing process, the semitrusted proxy cloud server cannot obtain any related information in cleartext. The cloud service provider uses the private key to decrypt the intermediate ciphertext to obtain the private data D.

$$\begin{aligned} &\mathrm{Decrypt}\left(\mathrm{PK}, C_{\mathrm{PK}_{u\longrightarrow d}}, \mathrm{SK}_d\right) \longrightarrow D, \\ &D = \frac{c_{pk_1} \cdot e\left(c_{pk_2}, A_2\right)}{C_{pk_1}^{A_1}}. \end{aligned} \tag{10}$$

6.10. Consensus Phase. The single consensus algorithm of the alliance chain cannot meet the environmental characteristics of low latency and high throughput in the Industrial Internet of Things environment. Combining the characteristics of PBFT [24] and Raft [25], a two-level mechanism is adopted to meet the environmental characteristics of the Industrial Internet of Things. The nodes are grouped, and the Raft consensus mechanism with supervisory nodes is used in the group, which has higher fault tolerance. The leadership committee elected by the Raft consensus mechanism uses the PBFT consensus mechanism, with reduced latency, improved throughput, and higher security. The specific process is shown in Figure 5. RBTF consensus process is as follows.

PBFT stage:

Step 1. After receiving client C's request, the master node (mian) will sort and sign the transactions and broadcast the prepacked message.

Step 2. After the secondary node (Replica) receives more than $2f$ messages, after verifying that the signature and other information are valid, it broadcasts a preparation message with an identity verification message.

Step 3. After receiving more than $2f + 1$ messages, the secondary node (Replica) judges whether the preparation phase is completed and enters the Raft consensus phase.

Raft stage:

Step 4. The leader in Raft broadcasts the message.

Step 5. After the follower nodes receive the message, they will verify the feedback.

Step 6. The leader node judges whether a consensus is reached according to the feedback result and submits the log.

Step 7. After completing the consensus, return the consensus result to the smart contract and write it into the blockchain ledger.

Related variables of RBFT:

(1) The RBFT consensus mechanism needs to meet the number of groups $k \geq 4$ and the number of nodes in the group $m \geq 3$

(2) The maximum fault tolerance range $f \leq \lfloor (k-1)/3 \rfloor$ in the PBFT phase and the maximum fault tolerance range in the Raft phase is $f \leq \lfloor (m-1)/2 \rfloor$

Table 2: Performance comparison between this article and other solutions.

Scheme	Not rely on trusted third parties	Content privacy	Identity privacy	Data validity	Verifiability	Traceability
[16]	√	√	√	×	−	√
[20]	√	√	−	×	√	×
[21]	√	×	×	√	−	×
[22]	√	√	√	×	√	×
[23]	×	√	−	×	√	−
This paper	×	√	√	√	√	√

7. Specific Structure

7.1. Security Proof

Lemma 1. *Based on the DBDH assumption, our scheme can resist selected plaintext attacks under the random oracle model; our solution is IND-CPA secure, that is, it satisfies the indistinguishability under the selected plaintext attack.*

Proof. Assuming that A is an adversary of arbitrary polynomial time, it can access the reencryption key oracle PKGen() and the reencryption oracle ReEnc(). The games of adversary A and challenger B are as follows.

Stage 1:

(1) Preparation stage: B randomly selects $a, b, c \in \mathbb{Z}_p^*$ to generate $(\mathrm{PK}, \mathrm{MSK}) \longleftarrow \mathrm{Setup}(1^\lambda)$

(2) Key generation: A initiates a key inquiry; B calculates $(pk_A, sk_A) \longleftarrow \mathrm{KeyGen}(\mathrm{MSK}, \mathrm{PK}, \mathrm{id}_A)$ and sends (pk_A, sk_A) to$A$

Stage 2:

Reencryption key generation: when A initiates a query for generating a reencryption key, according to the query information $(sk_{A'}, pk_{A'})$, B randomly selects $k_1, k_2 \in \mathbb{Z}_p^*$ to calculate (rk_1, rk_2) and returns $\mathrm{RK}_{A \longrightarrow A'} \longleftarrow \mathrm{RKGen}(\mathrm{PK}, sk_{A'}, pk_{A'})$ to A.

Reencryption: when A initiates a reencryption query, B returns $C_{\mathrm{PK}_{A \longrightarrow A'}} \longleftarrow \mathrm{ReEnc}(\mathrm{PK}_{A \longrightarrow A'}, c)$ and sends $C_{\mathrm{PK}_{A \longrightarrow A'}}$ to A.

Challenge: let $m_0, m_1 \in M$. A initiates a challenge and inquiry; B randomly selects a bit $\sigma \in (0, 1)$ and sends $C^* \longleftarrow \mathrm{Enc}(pk_{A'}, m_\sigma)$ to the opponent A. □

The opponent A outputs a guess value $\sigma' \in (0, 1)$; if $\sigma = \sigma'$, the opponent wins the game. The advantage of the adversary A can be defined as $\mathrm{Adv}_{\prod A}^{\mathrm{CPA}}(\kappa) = |\Pr[\sigma' = \sigma] - (1/2)|$. The probability that the adversary A guesses the correct σ' in the PPT is $(1 + \varepsilon)/2$, and ε is a nonnegligible quantity. The adversary A wins the game with at least the probability of $\varepsilon' = (1 + \varepsilon)/2 - 1/2 = \varepsilon/2$. Therefore, the advantage of the adversary A in the game can be ignored, and this solution is IND-CPA safe.

8. Performance Analysis

8.1. Content Privacy. In this solution, all private data is encrypted by the data owner using a sufficiently secure encryption algorithm and then uploaded to the proxy cloud server. We assume that the encryption algorithm used is sufficiently secure under the security model. If the key cannot be obtained, any internal adversary or external adversary cannot obtain the ciphertext. The data owner uses the private key of the cloud service provider to generate a reencryption key, which is reencrypted by the proxy cloud server and sent to the cloud service provider. Therefore, only authorized institutions can decrypt to obtain the ciphertext, and other entities in the process cannot obtain the ciphertext information.

8.2. Identity Privacy. When each participant entity registers, the identity certificate authority in the alliance chain will strictly examine the legality of the data owner or cloud service provider's identity and generate pseudoidentities for participants to ensure the privacy of their identities in transactions. In data sharing transactions, the real identity of the user cannot be obtained in the interaction between the participating entities and the smart contract. The cloud service provider only publishes the required keywords to achieve partial privacy protection and prevent the data owner from forging false data.

8.3. Data Validity. In this solution, only organizations authorized by the data owner can decrypt private data. The data owner generates a zero-knowledge proof π based on the private data. After submitting it to the smart contract, it can automatically verify whether the data meets the data requirements of the cloud service provider. Ensure the validity of the data.

8.4. Verifiability. The data owner sends the digital signature together with the generated zero-knowledge proof, and other entities can verify the validity of the signature, ensuring the validity of the zero-knowledge proof. The hash of the data is stored on the blockchain. The characteristics of the blockchain ensure that the data cannot be tampered with, and other entities receiving the data can verify the integrity of the data.

8.5. Traceability. In this solution, when the data owner and the cloud service provider reach a transaction on the premise that the data is complete and valid, the transaction is stored

in the blockchain. If the data owner fails to abide by the promise of no longer selling information to others and sells the information multiple times, the transaction history can be traced back in the blockchain to impose punishment.

9. Performance Analysis

Through comparative analysis of existing data sharing schemes, literature [20] uses time-based zero-knowledge proof and authentication encryption to perform mutual authentication between multiple attributes, ensuring security and privacy in data sharing. Literature [16] combined with zero-knowledge proof technology to achieve confidentiality and correctness in the data sharing process. Literature [21] realizes the reliability of data in data sharing among participants through an incentive mechanism, but the shared data is neither anonymous nor encrypted. Literature [22] realizes the sharing of vehicle data, but when there is an error in the data transaction, the traceability of the data source cannot be realized. Literature [23] uses proxy reencryption based on blockchain to store and share Internet of Things data in a cloud proxy server to achieve confidentiality and integrity in the data sharing process, but it cannot guarantee the validity and consistency of the data.

This article realizes the validity and consistency of the data sharing process under the premise of protecting data privacy and the privacy of the data owner's identity. Utilize the immutability and traceability of the blockchain to realize data integrity and traceability of data transactions. The comparison between this article and other programs is shown in Table 2.

10. Conclusions

Blockchain combined with zero-knowledge proof provides a new solution to the data sharing model. A large amount of data in the Industrial Internet of Things is the basis for promoting better development of services. How to maintain data privacy as much as possible on the premise of effective use of data is an important issue facing now. In response to these problems, this article combines zero-knowledge proof and smart contracts to achieve data validity and consistency between data owners and cloud service providers. Use proxy reencryption technology to realize the safe sharing of data among multiple participants. And combined with the nontamperable and traceable characteristics of the blockchain, the data can be verified and the transaction can be traced. Future work will study the realization of the secure sharing of data without a third-party server and the realization of a completely decentralized data sharing scheme.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Nos. 62162039 and 61762060) and the Foundation for the Key Research and Development Program of Gansu Province, China (Grant No. 20YF3GA016).

References

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of things: challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[2] Q. Li, B. Xia, H. Huang, Y. Zhang, and T. Zhang, "TRAC: traceable and revocable access control scheme for mHealth in 5G-enabled IIoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3437–3448, 2022.

[3] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial Internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3690–3700, 2017.

[4] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.

[5] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[6] E. B. Sasson, A. Chiesa, C. Garman et al., "Zerocash: decentralized anonymous payments from bitcoin," in *2014 IEEE symposium on security and privacy*, pp. 459–474, Berkeley, CA, USA, 2014.

[7] A. Rondelet and M. Zajac, *ZETH: On Integrating Zerocash on Ethereum*, 2019.

[8] J. Li, S. Wu, Y. Yang, F. Duan, H. Lu, and Y. Lu, "Controlled sharing mechanism of data based on the consortium blockchain," *Security and Communication Networks*, vol. 2021, 10 pages, 2021.

[9] F. Loukil, C. Ghedira-Guegan, K. Boukadi, and A. N. Benharkat, "Privacy-preserving IoT data aggregation based on blockchain and homomorphic encryption," *Sensors*, vol. 21, no. 7, p. 2452, 2021.

[10] X. Qin, Y. Huang, Z. Yang, and X. Li, "LBAC: a lightweight blockchain-based access control scheme for the Internet of things," *Information Sciences*, vol. 554, pp. 222–235, 2021.

[11] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.

[12] Y. Tian, T. Li, J. Xiong, M. Z. Bhuiyan, J. Ma, and C. Peng, "A blockchain-based machine learning framework for edge services in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, 2022.

[13] B. Muthusenthil, D. Nivetha, and H. Kim, "Reencryption scheme for secure data sharing," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1170–1174, Melmaruvathur, India, 2016.

[14] N. Mahakalkar and V. Sahare, "Implementation of reencryption based security mechanism to authenticate shared access in cloud computing," in *2017 International Conference*

on Trends in Electronics and Informatics (ICEI), pp. 547–550, Tirunelveli, India, 2017.

[15] Y. Fan, Y. Liao, F. Li, S. Zhou, and G. Zhang, "Identity-based auditing for shared cloud data with efficient and secure sensitive information hiding," *IEEE Access*, vol. 7, pp. 114246–114260, 2019.

[16] C. Huang, D. Liu, J. Ni, R. Lu, and X. Shen, "Achieving accountable and efficient data sharing in industrial Internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1416–1427, 2021.

[17] M. J. M. Chowdhury, A. Colman, M. A. Kabir, J. Han, and P. Sarda, "Blockchain as a notarization service for data sharing with personal data store," in *2018 17th ieee international conference on trust, security and privacy in computing and communications/12th ieee international conference on big data science and engineering (TrustCom/BigDataSE)*, pp. 1330–1335, New York, NY, USA, 2018.

[18] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2020.

[19] Y. Wang, A. Zhang, P. Zhang, and H. Wang, "Cloud-assisted EHR sharing with security and privacy preservation via consortium blockchain," *IEEE Access*, vol. 7, pp. 136704–136719, 2019.

[20] A. S. Sani, D. Yuan, W. Bao et al., "Xyreum: a high-performance and scalable blockchain for IIoT security and privacy," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1920–1930, Dallas, TX, USA, 2019.

[21] M. Shen, J. Duan, L. Zhu, J. Zhang, X. Du, and M. Guizani, "Blockchain-based incentives for secure and collaborative data sharing in multiple clouds," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1229–1241, 2020.

[22] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "An efficient and anonymous blockchain-based data sharing scheme for vehicular networks," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, Rennes, France, 2020.

[23] A. Manzoor, A. Braeken, S. S. Kanhere, M. Ylianttila, and M. Liyanage, "Proxy re-encryption enabled secure and anonymous IoT data sharing platform based on blockchain," *Journal of Network and Computer Applications*, vol. 176, article 102917, 2021.

[24] R. Kashyap, K. Arora, M. Sharma, and A. Aazam, "Security-aware GA based practical byzantine fault tolerance for permissioned blockchain," in *2019 4th International Conference on Control, Robotics and Cybernetics (CRC)*, Tokyo, Japan, 2019.

[25] L. Hou, X. Xu, K. Zheng, and X. Wang, "An intelligent transaction migration scheme for RAFT-based private blockchain in Internet of things applications," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2753–2757, 2021.