electronics

MDPI

Article

A Threshold Proxy Re-Encryption Scheme for Secure IoT Data Sharing Based on Blockchain

Yingwen Chen †, Bowen Hu *,†, Hujie Yu, Zhimin Duan and Junxin Huang

College of Computer, National University of Defense Technology, Changsha 410073, China; ywch@nudt.edu.cn (Y.C.); yhj258@nudt.edu.cn (H.Y.); zm_duan@163.com (Z.D.); huang_jx14@nudt.edu.cn (J.H.)
* Correspondence: hbw_14@nudt.edu.cn
† These authors contributed equally to this work.

Abstract: The IoT devices deployed in various application scenarios will generate massive data with immeasurable value every day. These data often contain the user's personal privacy information, so there is an imperative need to guarantee the reliability and security of IoT data sharing. We proposed a new encrypted data storing and sharing architecture by combining proxy re-encryption with blockchain technology. The consensus mechanism based on threshold proxy re-encryption eliminates dependence on the third-party central service providers. Multiple consensus nodes in the blockchain network act as proxy service nodes to re-encrypt data and combine converted ciphertext, and personal information will not be disclosed in the whole procedure. That eliminates the restrictions of using decentralized network to store and distribute private encrypted data safely. We implemented a lot of simulated experiments to evaluate the performance of the proposed framework. The results show that the proposed architecture can meet the extensive data access demands and increase a tolerable time latency. Our scheme is one of the essays to utilize the threshold proxy re-encryption and blockchain consensus algorithm to support IoT data sharing.

Keywords: IoT; blockchain; proxy re-encryption; data sharing

Citation: Chen, Y.; Hu, B.; Yu, H.; Duan, Z.; Huang, J. A Threshold Proxy Re-Encryption Scheme for Secure IoT Data Sharing Based on Blockchain. *Electronics* **2021**, *10*, 2359. https://doi.org/10.3390/electronics10192359

Academic Editors: Hung-Yu Chien, Chun-I Fan and Chunhua Su

Received: 30 July 2021
Accepted: 23 September 2021
Published: 27 September 2021

1. Introduction

With the rapid advancement of Internet of Things (IoT) technology, massive IoT devices have been deployed to different application scenarios. A fantastic amount of data is generated every day all over the world by these devices [1]. Data are the core concept of IoT technology, and these data are of inestimable value in different applications. Although IoT seems to be very attractive, its advances have brought new challenges to security and privacy. Consequently, there is an imperative need to guarantee the reliability and security of IoT data sharing [2].

IoT data sharing has played a vital role in smart cities, healthcare, vehicular networks and other application fields [3]. Figure 1 depicts a popular data sharing architecture. The data producer domain contains a series of sensors and other devices that can collect data directly. The data owners generate and manage a large amount of IoT data. Due to the limited storage capacity of IoT devices, these massive data need to be encrypted and uploaded to a third party for storage, which is convenient for data management, distribution and sharing. They can encrypt the data and upload them to the storage service providers, such as cloud servers and distributed file systems for storage. The ownership of the data belongs to the data owner domain. In order to facilitate the sharing of data with other users, access rights are usually bound to the encrypted data itself and outsourced to storage service providers for management. Therefore, it is necessary to establish an effective access control mechanism to ensure the privacy and security of data owners, but this task is usually quite challenging. Data sharing depends on semi-trusted storage service providers, and they may have incentives for trying to read the data. At the same time, the process of authentication and revocation of data access rights is also opaque to the data owners. Since

the data are encrypted, the users also lack an entirely credible communication channel to share the decryption key. Moreover, the data sharing depends on storage service providers, which will bring excellent communication overload. If the data owners want to share data with the receivers directly, they need to download the data, decrypt and re-encrypt them for data sharing. This calculation and communication overload is enormous. However, the computing power of the data owners is limited, and they cannot afford to decrypt the encrypted data and then encrypt it to the receivers. Therefore, they can share IoT data with the help of the proxy re-encryption mechanism. In this way, the task of the data owners can be changed from the encryption and decryption process of ciphertext to the generation process of a re-encryption key, which distributes the overload on the data owners.

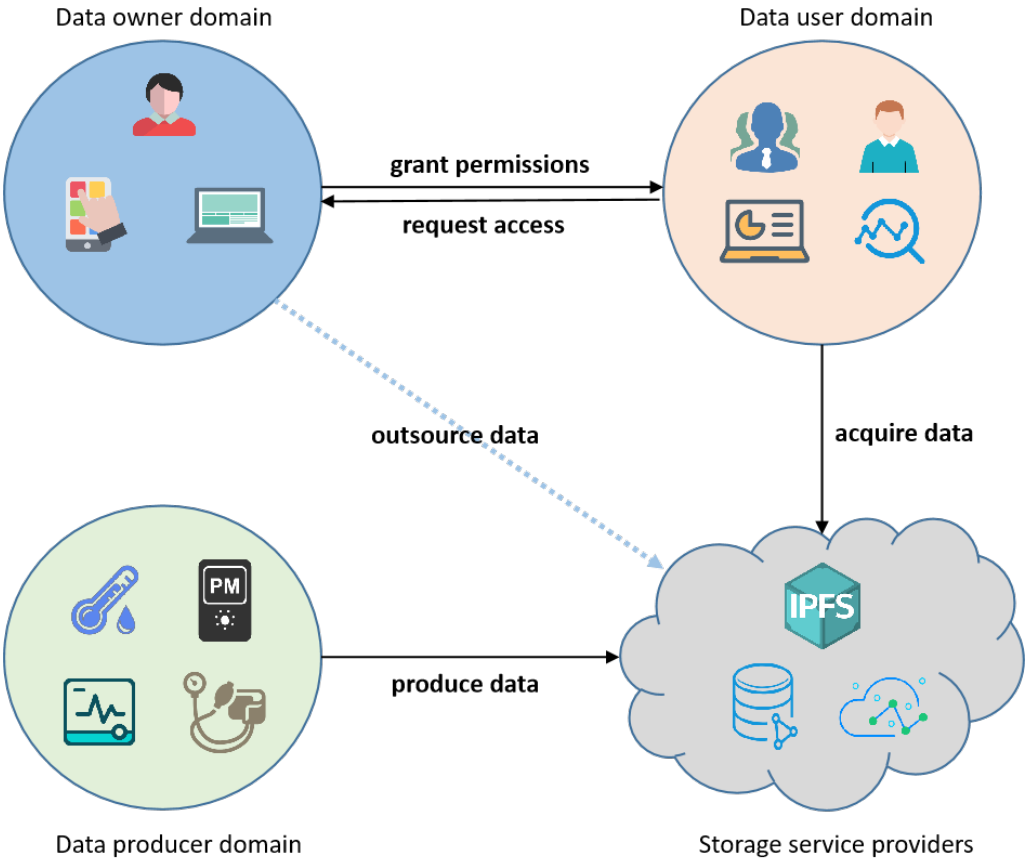

Figure 1. The traditional data-sharing architecture.

The concept of proxy re-encryption was first proposed by Blaze, Bleumer, and Strauss [4]. It is considered to be an effective cryptographic tool for encrypted data sharing. As shown in Figure 2, in this mechanism, a semi-trusted proxy converts Alice's (delegator) ciphertext to Bob (delegatee), which can be decrypted with Bob's private key [5]. In the process of sharing, the data plaintext is not revealed, and the proxy does not need to obtain the private key of either party. However, the single semi-trusted proxy may have malicious behavior in the process of ciphertext conversion. The first risk of this mechanism is collusion between the proxy and the delegatee. Through such illegal means, the delegatee can decrypt data that they do not have permission to access without the consent of the delegator. Another risk is that the single proxy is vulnerable to a denial of service attack. Furthermore, the single proxy may have both a performance bottleneck and a single point of failure.

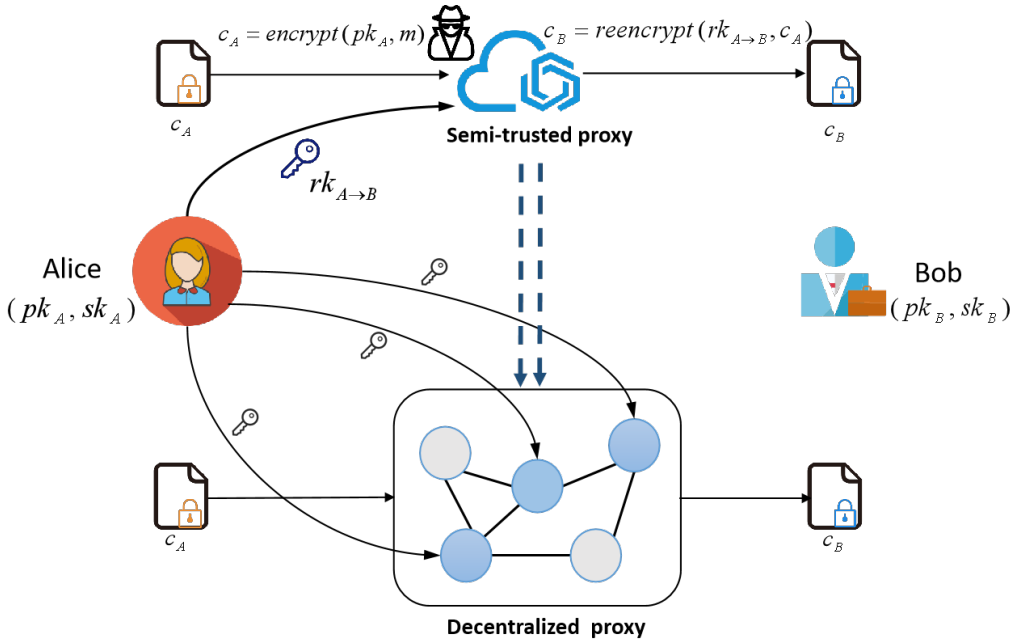

Figure 2. The comparison between third-party proxy servers and decentralized proxy servers.

Threshold proxy re-encryption delegate re-encryption to multiple proxy nodes [6]. This mechanism can split the trust between multiple proxies, which can effectively prevent third-party proxy servers from converting unauthorized ciphertexts. When a proxy node carrying out re-encryption calculation paralyzes or behaves maliciously, the threshold proxy re-encryption mechanism can provide a secure and reliable service to the delegator. Blockchain technology has the characteristics of decentralization, untamperability and data traceability, which can be used to build a transparent, open, secure and reliable data-sharing environment, and can be used to control, verify and audit third-party access to data. Because the blockchain technology has inherent security and identification [7], the access request records of data users are untampered in the blockchain ledger.

The threshold proxy re-encryption requires the support of multiple proxy nodes, and there are a large number of decentralized nodes with high computing power in blockchain networks. Usually, the computing power of these nodes is consumed in calculating a mathematical problem to complete the block mining process, such as Bitcoin [8]. Our idea is to combine the consensus algorithm mechanism of blockchain with re-encryption computing. As shown in Figure 2, the data delegator splits the re-encryption key and sends it to multiple proxy nodes, which can be selected from the consensus nodes in the blockchain network. After receiving the re-encryption request from the delegator, these nodes calculate and generate the corresponding "re-encryption shares" according to the received re-encryption key, and record the request and the ciphertext conversion in the blockchain ledger. This mechanism not only ensures the reliability and security of the re-encryption process, but also makes full use of the surplus computing power in the blockchain network. In addition, the token incentive mechanism attached to the blockchain can be used to encourage the re-encryption behavior of the proxy nodes.

In this paper, a new encrypted data-storing and -sharing architecture is proposed by combining threshold proxy re-encryption with blockchain technology. The traditional proxy re-encryption mechanism relies on semi-trusted third-party service providers to complete data sharing. We design a consensus mechanism based on threshold proxy re-encryption eliminates dependence on the third-party central service providers. Multiple consensus nodes in the blockchain network act as proxy service nodes to re-encrypt data and combine converted ciphertext. This mechanism can ensure the reliability, security and auditability of IoT data sharing.

The contributions of our work are as follows. (i) Our system uses a decentralized network to remove the reliance on central proxy service providers, utilizing security features of the blockchain to ensure reliability, availability, and correctness. (ii) It combines the re-encryption process of threshold proxy re-encryption with the consensus algorithm mechanism in blockchain networks, and effectively utilizes the rich computing resources in the blockchain network to improve the performance and security of the ciphertext conversion process. (iii) Our method implements a decentralized key management system in which the symmetric key that users use to encrypt data arenot disclosed in the network. This mechanism solves the limitation of using consensus networks to store and share private encrypted data securely and supports sharing sensitive data for distributed and centralized applications.

2. Related Work

Considerable research works have been carried out to solve the problem of IoT data sharing. In this section, we review some typical work of data sharing and the development of the main technologies involved in this paper.

2.1. Data Sharing Based on Third-Party Service Providers

Many data-sharing architectures have been proposed for the data-sharing environment based on third-party service providers, such as cloud computing and edge computing [9]. The cloud-assisted IoT [10] framework is proposed to solve massive data mutuality between the cloud and IoT and adopt a conditional identity-based broadcast proxy re-encryption scheme (CIBPRE) to realize secure data sharing. Under this framework, users can confidentially collect, store, and share IoT data. Yin et al. [11] propose an attribute-based encryption with keyword search (ABKS) scheme, which can achieve fine-grained and efficient data access control over encrypted data. This scheme is very suitable for a secure data-sharing environment, such as the currently popular cloud computing. However, the third-party entity service providers may have malicious behavior. The researchers of [12] design a novel certificateless hybrid signcryption scheme without pairing. That provides an efficient and simple key management mechanism. Since the private key generator (PKG) does not generate a complete private key for users, this method can well resist collusion attacks. The edge computing architecture is introduced to solve the delay problem of data sharing in cloud computing, but it also brings problems in data privacy protection. Cui et al. [13] design a proxy-aided ciphertext-policy attribute-based encryption (PA-CPABE) scheme. In the process of data sharing, the distribution of the data decryption key does not require a secure channel. Moreover, this architecture can effectively improve the efficiency of data sharing by outsourcing decryption operations to edge devices. This scheme can provide scalable access control to support data security and privacy in edge computing.

2.2. Blockchain-Based Data Sharing and Access Control Scheme

By deploying the Ethereum blockchain network as distributed middle-ware [14], the authors propose an IoT architecture that uses edge computing for data sharing. The framework supports data producers at the edge granting access to their data to different third parties. Leveraging the immutability properties of blockchain and the distributed nature of smart contracts, data owners can audit and be aware of the whole process of data sharing. This method can ensure data ownership and privacy security. Jiang et al. [15] present a compelling data-sharing scheme based on blockchain for vehicular social networks. This scheme establishes trust relationships between vehicles in the blockchain network by combining the pseudonym generation mechanism and the identity-based signature mechanism. Moreover, this scheme utilizes the consensus mechanism to ensure the traceability and confidentiality of data sharing. Xia et al. [16] establish a medical data-sharing system based on blockchain. The smart contracts and access control mechanism are adopted to track the behavior of data sharing effectively. When the violation of the data requester is detected, the access to the violating entity is revoked. This scheme can effectively provide access

control, data provenance and auditing for medical data sharing. The authors [17] propose a proxy re-encryption scheme based on blockchain. It encrypts the IoT data and stores them in the distributed cloud. This system designs a dynamic smart contract between data owners and data users for data sharing without the participation of a trusted third party. The shared data are only visible by the owner and the person present in the smart contract by using a very efficient proxy re-encryption scheme. This novel combination of smart contracts with proxy re-encryption provides a secure platform for the storing and sharing of IoT data.

2.3. Data Sharing Based on Proxy Re-Encryption Mechanism

Much research work has been carried out around the data-sharing scheme based on the proxy re-encryption mechanism. Ateniese et al. [18] propose a new PRE (proxy re-encryption) scheme. This scheme is the first uni-directional PRE scheme based on bilinear mapping. It can be applied to secure distributed storage, which shows that the PRE scheme can be used as an effective method to implement secure access control and encrypted data sharing in the file system. On this basis, ref. [19] design an identity-based proxy re-encryption (IB-PRE) scheme by combining the PRE scheme with the IBE (identity-based encryption) scheme. In this scheme, the proxy server can directly convert the encrypted ciphertext based on the identity information of user A into the encrypted ciphertext based on the identity information of user B. In view of the fact that data confidentiality in unreliable storage cannot be guaranteed, the authors [20] propose a new identity-based proxy re-encryption algorithm and apply it to the access control scheme of cloud storage. The re-encryption key is divided into two parts: one for re-encryption and the other for authorization. In this scheme, there is no need to completely regenerate the re-encryption key for different users, only part of the calculation is needed, and the computational complexity is low. Shao et al. combine the conditional PRE scheme into IB-PRE [21] and propose the first conditional IB-PRE scheme (IB-CPRE). This scheme allows a proxy to transform the ciphertexts under an identity to other ciphertexts under another identity. The condition means that the cryptographer can embed the conditional parameters ω when encrypting the ciphertext, and the ciphertext can be re-encrypted if and only if the proxy holds a conversion key embedded with the same conditional parameters ω, so the re-encryption behavior of the proxy is limited. Reference [22] combine keyword search in public-key encryption with attribute proxy re-encryption, design an attribute proxy re-encryption scheme supporting keyword search, and prove its security under the random prophecy machine model. Using proxy re-encryption for data authorization and sharing is an effective technology that can be applied to various application scenarios of data sharing.

The main goal of these studies is to improve the data encryption process to realize the proxy re-encryption mechanism with various characteristics. The focus of our framework is to improve the ciphertext conversion process of proxy re-encryption. Our scheme combines the consensus mechanism of blockchain with threshold proxy re-encryption. On the one hand, it takes advantage of decentralization to transform centralized proxy servers into multiple decentralized consensus nodes. On the other hand, it uses surplus computing power in the blockchain network to complete the ciphertext conversion process, which improves the reliability and performance of ciphertext conversion in the proxy re-encryption mechanism. Those are the crucial aspects that other studies have not paid attention to.

3. Background

3.1. Bilinear Maps

Our threshold proxy re-encryption algorithm is built on bilinear maps. The definition of bilinear maps is described as follows: there are two cyclic groups G_1 and G_2 of order p, and there is a bilinear map e between them: $G_1 * G_1 \rightarrow G_2$. The map satisfies the following three properties:

(1) Computability: given $g_1, g_2 \in G_1$ there is a polynomial time algorithms to compute $e(g_1, g_2) \in G_2$.

(2) Bilinearity: for any integers $x, y \in [1, p], e(g^x, g^y) = e(g, g)^{xy}$.
(3) Non-degeneracy: if g is a generator of G_1 then $e(g, g)$ is a generator of G_2. In other words, this can be simplified as $e(g, g) \neq 1$.

The size of G_1, G_2 is determined by the security parameter.

3.2. Threshold Secret Sharing

The (t, n) secret sharing scheme is a method of sharing secrets. It divides secrets and distributes them to N participants for holding. Secrets can only be recovered when at least t participants are present. The most typical scheme is Shamir's secret sharing scheme [23]:

There is a group of n participants $(p_1, p_2, \ldots, p_n)$. The delegator randomly picks a polynomial $f(x) = s + \sum_{j=1}^{k-1} a_j x^j$ of $(k-1)$ degrees, where s is a constant in $f(x)$. Each participant p_i assigned a unique domain element b_i and receives the corresponding secret share $s_i = f(b_i)$ through the privacy channel. If at least t participants in the group $(p_1, p_2, \ldots, p_n)$ are present, the participants can recover secrets by s:

$$f(x) = \sum_{j=1}^{k} f(b_j)\lambda_{ij} \tag{1}$$

where $\lambda_{ij} = \prod_{l=1}^{k} \frac{x - b_l}{b_i - b_l}, l \neq i$, and $s = f(0)$.

If fewer than t participants are present, the participants cannot recover secrets because they cannot derive any information about the polynomial $f(x)$.

3.3. Blockchain

Blockchain is a cross-disciplinary discipline formed by integrating mathematics, cryptography and computer science in which a large number of information security technologies are applied. Unlike the traditional database security mechanism, blockchain allows any node to join the network, and all its ledger data are open. Blockchain is an ever-increasing shared distributed ledger that can prevent tampering.

The member nodes in the network share, copy and synchronize the distributed ledger. The distributed ledger is recorded and permanently stored in a continuous chain of encrypted hash blocks. Without the participation of central authorities or trusted third parties, participating nodes in the blockchain network can update records in the distributed ledger and reach consensus. Unless all network members reach an agreement in subsequent transactions, the ledger may not be tampered or revoked.

3.4. Consensus Algorithm

The consensus mechanism is a core component of the blockchain network. Consensus refers to the result that multiple participants who are independent of each other agree on a specific issue. The consensus in the blockchain refers to the consistency reached by each node to a certain block in an open decentralized network. The consensus mechanism mainly studies the allocation of accounting rights generated by blocks and the verification after the emergence of blocks. At present, around the research content of consensus mechanism, there are mainly consensus algorithms in the blockchain system, such as Proof of Work (POW) [24], Proof of Stake (POS) [25], Delegated Proof of Stake (DPOS) [26] and Practical Byzantine Fault Tolerance (PBFT) [27], etc.

According to the way new blocks are generated, the consensus mechanism can be divided into proof, Byzantine and DAG (Direct Acyclic Graph) classes. POW is the consensus mechanism of workload proof. The consensus mechanism was first applied in Bitcoin. In this consensus mechanism, each node performs a hash operation to compete for the block accounting rights, and after the block is generated, it is broadcast in the whole network for other nodes to verify it. However, the generation of blocks consumes a lot of computing power and other resources, and it takes a long time to achieve data consistency. According to the consensus idea of POW, in our architecture, the process of calculating

the hash problem of consensus nodes is replaced by the process of converting ciphertext by using re-encryption keys, and new blocks are generated and written, according to the verification results of other nodes.

4. Proposed Method

4.1. System Framework and Workflow

Our proposed method provides a new solution for the encrypted data sharing of the IoT. Figure 3 provides an overview of the data-sharing architecture. In this section, we will explain the participating entities and the workflow design of the data-sharing architecture in detail.

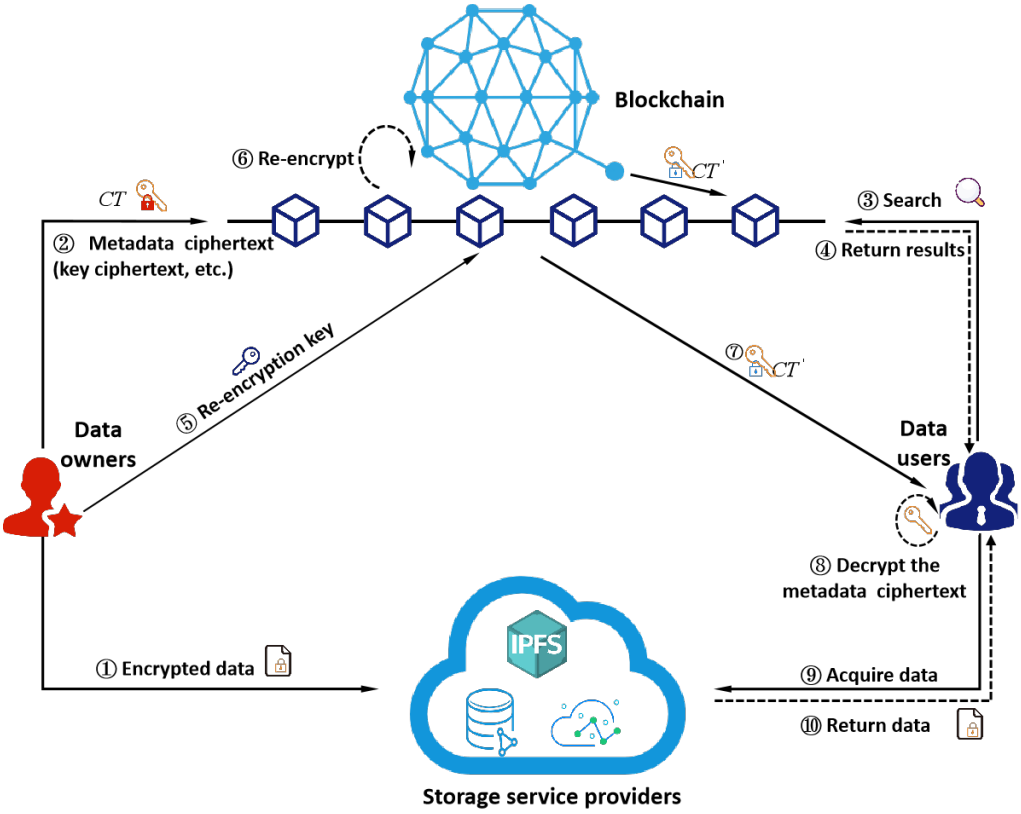

Figure 3. The overview of our data-sharing architecture.

As described in Figure 3, our system mainly consists of four entities: data owners (DO), data users (DU), storage service providers (SSP), and the blockchain network.

Data owners (DO): DO have ownership of the personal data, and the IoT devices to which they belong will collect and generate a large amount of data. These data usually involve sensitive personal information. Therefore, encryption processing is required before being uploaded to the far end for storage. In order to save the time and computational overload of data sharing, The data itself are encrypted with a symmetric key and stored in SSP. Meanwhile, DO encrypt the metadata information of the data with their own public key (asymmetric encryption). Moreover, the metadata ciphertext is posted to the blockchain for storing, including the data summary, data store location, the signature of the data ciphertext and the symmetric key. When receiving an access request from data users, DO verify their access rights and generate re-encryption keys based on DU's public key and their own private key. The re-encryption key will be sent to the consensus nodes of the blockchain for ciphertext conversion and sharing.

Data users (DU): include a series of institutions and applications that need to use the IoT devices' data. DU search for the content of interest by browsing the data summary on the blockchain. In our previous research work, we have designed a smart contract frame-

work based on the attribute-based access control (ABAC) and the searchable encryption algorithm [28]. In this architecture, the role of blockchain is not limited to access control, which can achieve precise and fine-grained access control. Moreover, it can also achieve efficient and privacy-protected data retrieval. Data users can search for encrypted keywords stored on the blockchain through keywords to obtain corresponding metadata. Because the focus of this paper is not on the access control mechanism, it will not be discussed in detail. After obtaining the authorization of the data owners, the blockchain node generates a corresponding re-encrypted metadata ciphertext, and the DU can use their private key to decrypt the re-encrypted ciphertext to obtain metadata information. According to the data storage location, the DU send access requests to SSP to acquire encrypted data, and the plaintext of the data can be obtained by decrypting the ciphertext with the symmetric key. Furthermore, The integrity and correctness of data can be verified by data summary and signature.

Blockchain Network: The blockchain network is the critical component of this system. The distribution and sharing of encrypted symmetric key ciphertext depend on the proxy re-encryption mechanism, in which the proxy servers play a vital role. In our system, the blockchain consensus node plays the role of proxy servers. The threshold proxy re-encryption scheme is adopted to give better play to the advantage of the decentralization of the blockchain network. The DO distribute the multiple shares of re-encryption keys to multiple nodes in the blockchain network, respectively. Consensus nodes using the corresponding re-encryption share to re-encrypt the original ciphertext. These converted metadata ciphertexts are collected and packaged into new blocks. In this process, the private key of the data owners and ciphertext information is not leaked into the network. The implementation of this process depends on the blockchain consensus algorithm we designed, which is described in detail in the subsequent chapters of this paper. The DU can read the corresponding metadata ciphertext in the blockchain ledger. By replacing the third-party semi-honest agent in the proxy re-encryption mechanism with a decentralized blockchain network, our method enables reliable and secure encrypted data sharing. All the ciphertext conversion records and data-sharing processes are recorded in the blockchain ledger immutably. According to those records, some proxy nodes' malicious access behaviors can be identified easily, which can improve system security.

Storage Service Providers (SSP): The data generated by the IoT devices usually are very massive. Those data collected by IoT devices must be encrypted and uploaded to third-party servers for storing and sharing. Cloud computing and edge computing are widely adopted to store and distribute IoT data. The distributed file system is also a very suitable storage method for storing massive IoT data, such as IPFS. Moreover, the distributed storage service providers can be well combined with a decentralized blockchain network, which can improve the security and reliability of encrypted data storage and sharing. In our system, these storage service providers are responsible for storing the user's encrypted IoT data. SSP cannot snoop the plaintext information of the data in the process of data sharing. By separating the grant and authentication of data access from the data storage service, Our framework provides a proper way to avoid privacy disclosure to malicious entities.

4.2. System Construction

We propose to combine the blockchain consensus algorithm and threshold proxy re-encryption mechanism, which consists of eight polynomial time algorithms defined as follows. The threshold proxy re-encryption mechanism is designed by referring to ECIES-KEM [29], and the BBS98 proxy re-encryption scheme [4]. This mechanism can perform proxy re-encryption process through a group of N blockchain consensus nodes. When at least t (out of n) of these nodes participate in performing re-encryption, the delegatee can obtain the converted new ciphertext in the blockchain network, and can use their private keys for decryption. In this section, we explain each of these phases in more detail.

- ***System setup*** $(1^\lambda, \mathbf{U}) \rightarrow PP$: Selecting a security parameter 1^λ as input, this algorithm outputs the public parameters PP. The users can use the PP to generate their own public and private key pairs. The generation process of PP is as follows.
  This setup algorithm takes a security parameter λ as the input and outputs a bilinear maps e : $G_1 \times G_1 \rightarrow G_2$, where G_1, G_2 is a multiplicative cyclic group of prime order q. Select a random generator $g \in G_1$, then calculate $Z = e(g, g)$. Define four hash functions: $H_0 : \{0,1\}^* \rightarrow G_1$, $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : G_2 \rightarrow \{0,1\}^{\log_2 q}$, $H_3 : G_2 \rightarrow \{0,1\}^*$.
  Run the parameter generation algorithm to generate common parameters:

$$Params = (g, Z, e, q, G_1, G_2, H_0, H_1, H_2, H_3), \tag{2}$$

- ***Key Generation*** $(PP) \rightarrow (SK_{DO}, PK_{DO}, SK_{DU}, PK_{DU})$: According to the generated public parameters, the DO and DU generate their own public and private key pairs, respectively. The DO randomly choose $a \in Z_q^*$ as their private key and calculate the public key as $PK_{DO} = g^a$. DU randomly choose $b \in Z_q^*$ as their private key and calculate the public key as $PK_{DU} = g^b$.
- ***Data Encryption*** $(DEK, Data) \rightarrow CT$: We use a symmetrical encryption mechanism to encrypt the IoT data. Symmetric key encryption requires users to know a common secret key. For convenience, we refer to this common secret key as DEK (data encryption key). The most useful symmetric key encryption algorithm for our purposes is AES [30], because it is normally hardware-accelerated, which can ensure the efficiency of data encryption and decryption. The encrypted data CT are uploaded to SSP for storage.

$$CT = encrypt(DEK, Data), \tag{3}$$

- ***Metadata Encryption*** $(PP, PK_{DO}, M) \rightarrow CT_M$: The data owners (DO) encrypt the metadata M with their public key, and then send the CT_M to the blockchain for storage and management. In this case, the data can only be decrypted by the DO private key. Metadata M includes the data decryption key DEK, data summary, data store location, the signature of DO about CT, etc.

$$CT_M = \mathrm{Enc}(PK_{DO}, M) = \left(g^{ak}, MZ^k\right), \tag{4}$$

  where $Z = e(g, g)$, k is a random coefficient
- ***Re-encryption Key Generation***: The data owners (DO) calculate and generate the re-encryption key according to their private key and the public key of the requester DU who want to acquire the data.
  On input of the secret key SK_{DO}, the public key of the intended delegatee PK_{DU}, the number of proxy nodes N, and the threshold t, the re-encryption key generation algorithm ReKeyGen computes N fragments of the re-encryption key between DO and DU.

$$RK_{DO \rightarrow DU} = g^{b/a} \tag{5}$$

  According to the following formula,

$$f(x) = \sum_{i=1}^{k} f(x_i)\lambda_{ij}, \lambda_{ij} = \prod_{l=1, l \neq j}^{k} \frac{x - x_l}{x_i - x_l}, \tag{6}$$

the re-encryption key share held by each proxy node is $rk^i_{O->U} = g^{f(x_i)} \bmod q$, $i = 1, 2, \ldots, n$, for proxies P_i. The DO will publish the ciphertext conversion requests to the blockchain network. The consensus nodes which are selected by specific rules, receive the re-encryption key fragments. These consensus nodes only hold a segment of the re-encryption key, and no node holds the complete re-encryption key.

- ***Re-encryption***: After the consensus nodes in the blockchain network receives the re-encryption key published by the DO, they begin to perform the ciphertext conversion process in response to re-encryption requests.
  This algorithm re-encrypts CT_M from the data owners to the data users, according to $CT_M = \left(g^{ak}, MZ^k\right)$, proxy nodes can convert the original ciphertext with the re-encryption key $rk^i_{O->U}, i = 1, 2, \ldots, n$ (for proxies P_i) to the CT'_M by calculating the following:

$$e\left(g^{ak}, \left(rk_{p_i}\right)^{\lambda_i}\right) = e\left(g^{ak}, g^{[\lambda_i \cdot f(x_i)]}\right) = Z^{\mathrm{ak}[\lambda_i \cdot f(x_i)]} \tag{7}$$

  every proxy nodes P_i can obtain the ciphertext:

$$C_{b_i} = \left(Z^{ak[\lambda_i f(x_i)]}, MZ^K\right) \tag{8}$$

  When t out of n proxy nodes correctly completes the re-encryption calculation, the ciphertext C_{b_i} can be combined to the new ciphertext CT'_M, which can be decrypted by DU with their own private key:

$$\prod_{i=1}^{t} Z^{ak[\lambda_i f(x_i)]} = Z^{ak \sum_{i=1}^{k} \lambda_i f(x_i)} = Z^{ak \cdot b/a} = Z^{bk} \tag{9}$$

$$CT'_M = \left(Z^{bk}, MZ^k\right) (\mathrm{mod} q) \tag{10}$$

  The combined converted ciphertext will be written into the new block after the proxy nodes complete the consensus confirmation.
- ***Metadata Decryption***: The DU can use their private key SK_{DU} to decrypt the metadata ciphertext $CT'_M = \left(Z^{bk}, MZ^k\right) = (\alpha, \beta)$ and the following metadata:

$$M = \beta / \alpha^{1/SK_{DU}} \tag{11}$$

- ***Data Decryption***: When the DU obtain the metadata M, they can obtain the data symmetric decryption key DEK and the data storage location. They can access SSP to obtain data plaintext and verify the integrity and correctness of the data through data summary and signature, which are included in the metadata.

$$Data = decrypt(DEK, CT) \tag{12}$$

4.3. Consensus Mechanism Based on Threshold Proxy Re-Encryption

In our system, the consensus mechanism is the core component. Compared with the traditional proxy re-encryption mechanism, our solution utilizes a decentralized network to eliminate dependence on third-party central service providers. The consensus nodes in blockchain networks jointly participate in the consensus process based on re-encrypted ciphertext conversion. The threshold proxy re-encryption can be perfectly combined with the consensus algorithm by splitting and distributing the re-encryption key to the consensus node in the blockchain network. Figure 4 shows an overview of the consensus algorithm based on threshold proxy re-encryption, and the following gives a detailed introduction to the consensus mechanism.

1. The data owners broadcast the generated re-encryption keys to multiple consensus nodes, and simultaneously send the requirements for ciphertext conversion to the blockchain network. The ciphertext conversion process requires a relatively large computational overload. Not all nodes in the network have re-encryption capability, so nodes with specific computational capability perform re-encryption calculation in response to this requirement, and these nodes act as consensus nodes in the blockchain network.
2. As shown in Figure 5, each consensus node performs re-encryption conversion on metadata ciphertext, according to the received re-encryption key. The consensus node first locates the corresponding original ciphertext CT_M in the blockchain ledger, according to the re-encryption requirements, and then uses the re-encryption key $rk^i_{O->U}, i = 1, 2, \ldots, n$ to perform the conversion process.
3. The consensus nodes that complete re-encryption calculating in a limited time sign the re-encrypted ciphertext $C_{b_i}, i = 1, 2, \ldots, n$ and broadcast them to the blockchain network;
4. Each consensus node collects and verifies the received re-encrypted ciphertext C_{b_i}. When a consensus node takes the lead in collecting t verified re-encrypted ciphertexts; it sends a request to become the leader node to the blockchain network and initiates a round of voting. After receiving this request, other consensus nodes verify the correctness of the t re-encrypted ciphertext. If more than half of the nodes support the result, the node that sent this request becomes the leader node and obtains the record right of this round of consensus process.
5. The leader node combines those re-encrypted ciphertext C_{b_i} to the new ciphertext CT'_M. Moreover, this leader node will add the new block written with re-encrypted records and converted ciphertext to the blockchain ledger and broadcasts it to the network. Other nodes complete consensus confirmation and update the ledger. Then, the threshold proxy re-encryption process is completed, and the converted metadata ciphertext CT'_M is written to the update block. The data users can obtain the converted metadata ciphertext from the blockchain ledger and complete the decryption.

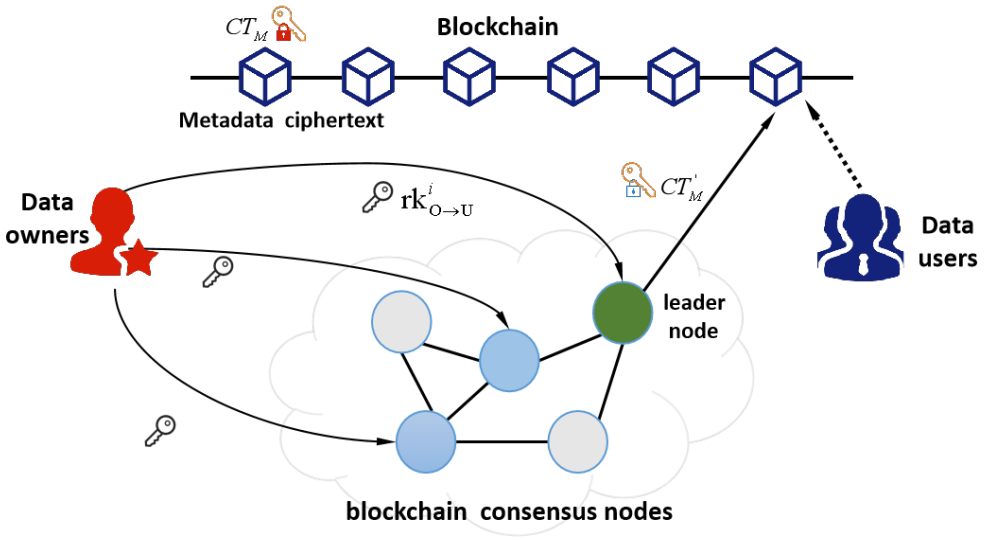

Figure 4. The consensus mechanism based on threshold proxy re-encryption.

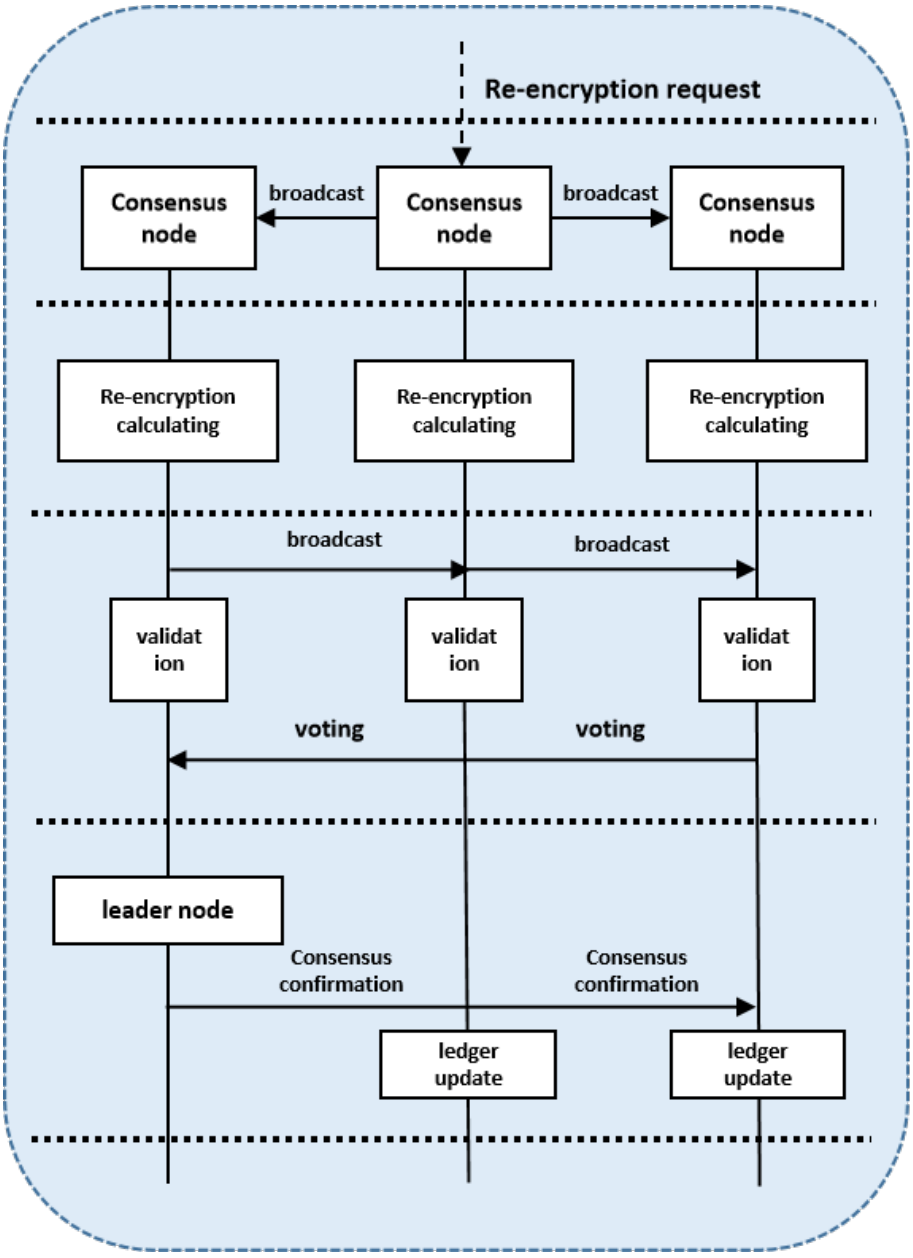

Figure 5. The workflow of data re-encryption processing and consensus confirmation.

5. Evaluation

5.1. Security Analysis

In this subsection, we analyze the typical attack model that our proposed architecture can prevent.

Collusion attack: The architecture we designed can well defend against collusion attacks. This threat model refers to the collusion between the data requester and the proxy service providers. In the traditional proxy re-encryption mechanism, malicious users can collude with dishonest proxy servers to recover the private key information of data owners. Our scheme utilizes Shamir's secret sharing algorithm to publish the re-encryption key to the consensus nodes of the blockchain network. The re-encryption key is divided into N parts and distributed to N consensus nodes, respectively. These consensus nodes only hold a segment of the re-encryption key, and no node holds the complete re-encryption key. Therefore, the collusion attack needs t consensus nodes and data receivers to complete together. Moreover, there is no centralized third-party proxy server, and decentralized consensus nodes have no motive for collusion. That makes it more difficult for consensus

nodes to collude with data receivers, so collusion attacks can be effectively prevented under realistic conditions.

Man-in-the-Middle attack: In the process of data sharing, there may be a middleman trying to decrypt the data for more profit. In our architecture, the data are encrypted and uploaded to the storage service providers. Furthermore, the decryption key and metadata information are encrypted again and stored in the blockchain network. So the plaintext information of the data is not exposed in the sharing process. Moreover, in the process of ciphertext conversion, each consensus node only holds the re-encryption key fragment. They cannot snoop any information about the plaintext of metadata and the private key of the data owners. The confidentiality of data can be effectively protected by our system. Moreover, the distributed design eliminates the threat of single point of failure.

Data tampering: Data users must acquire the metadata, including the data summary, before querying the complete data in the storage service providers. When the data stored in SSP are maliciously tampered with, the data users can detect that the data have been modified by recalculating the hash summary of the data. The metadata information of the data is stored in the blockchain distributed ledger. Any malicious node cannot tamper with it because each node has a complete replica. Meanwhile, the ciphertext conversion records of the consensus nodes are recorded in the blockchain ledger without tampering. The data owners can easily trace and audit the data-sharing process. Our architecture can ensure that the integrity of the data is not compromised.

5.2. Efficiency Analysis

This subsection provides a comprehensive description on the efficiency of our proposed architecture. A set of experiments were designed to analysis the performance of our system.

Our experimental environment was a Windows operating system desktop computer with 2.9 GHz, Intel i7, 16 GB RAM, 3200 MHz DDR4 specifications. We built a threshold proxy re-encryption scheme based on our designed blockchain simulation environment.

In order to evaluate the time cost of our introducing threshold proxy re-encryption mechanism for data sharing, we have measured the time overload of each stage of the proxy re-encryption mechanism for many times. The threshold proxy re-encryption mechanism can be divided into five stages: key generation, encryption, re-encryption key generation, data re-encryption and decryption. As shown in Figure 6, we can find that the data re-encryption stage takes significantly more time than the rest of the stages, with an average of 98 ms. That is because our data re-encryption process relies on the blockchain consensus mechanism, which takes quite a long time to complete consensus confirmation to ensure data consistency. Compared with the reliability and security of decentralization brought by this mechanism, we believe that this delay is acceptable. The remaining stages are completed at the user side, and the average time overload is less than 25 ms. This load is affordable for users.

The performance of threshold proxy re-encryption is closely related to two parameters: shares and threshold. In the re-encryption key generation stage, the data owners generate n shares re-encryption keys and send them to n blockchain consensus nodes. These consensus nodes re-encrypt the ciphertext, according to their received re-encryption keys, and generate n converted ciphertexts. When t out of n proxy nodes correctly complete the re-encryption calculation, the ciphertext can be combined to the new ciphertext. The threshold value t represents the minimum shares required to combine the converted ciphertext into a new decipherable ciphertext. Figure 7 shows that the more the split shares of the re-encryption key, the longer time the data re-encryption phase will take. Furthermore, with the increase of threshold, the time cost of the re-encryption process also increases slightly. This result shows that the efficiency of the consensus mechanism based on threshold proxy re-encryption is affected by these two re-key parameters. The more consensus nodes act as proxy server nodes and the more converted ciphertexts need to combine new ciphertexts, the longer the consensus confirmation time will be, and the more time it will

take to complete data re-encryption. The reliability and safety of the data-sharing system will also be improved. Therefore, the re-key parameter setting must take into account the two aspects of time overload and safety.

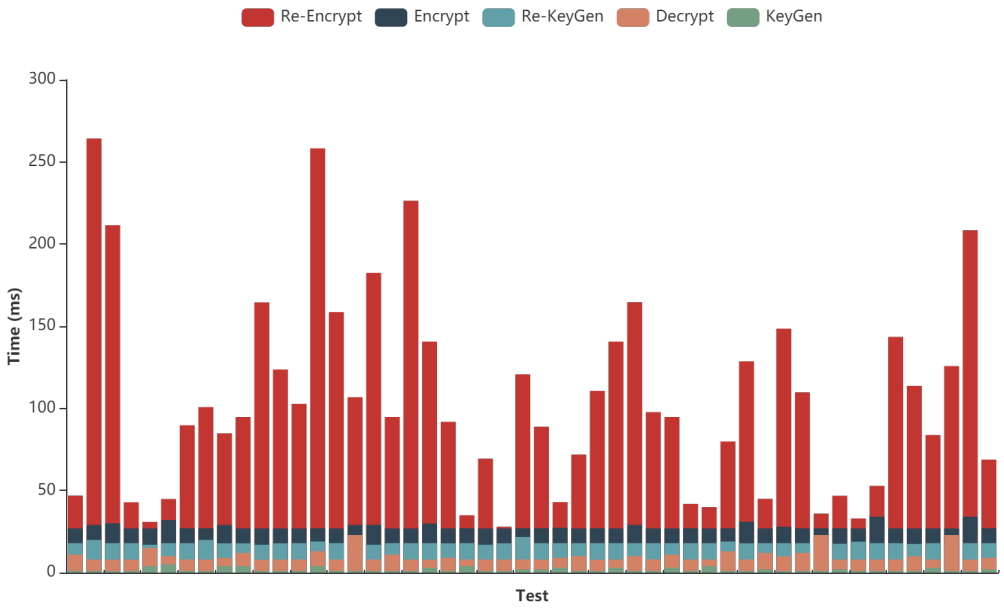

Figure 6. The time consumption in different stages of data sharing.

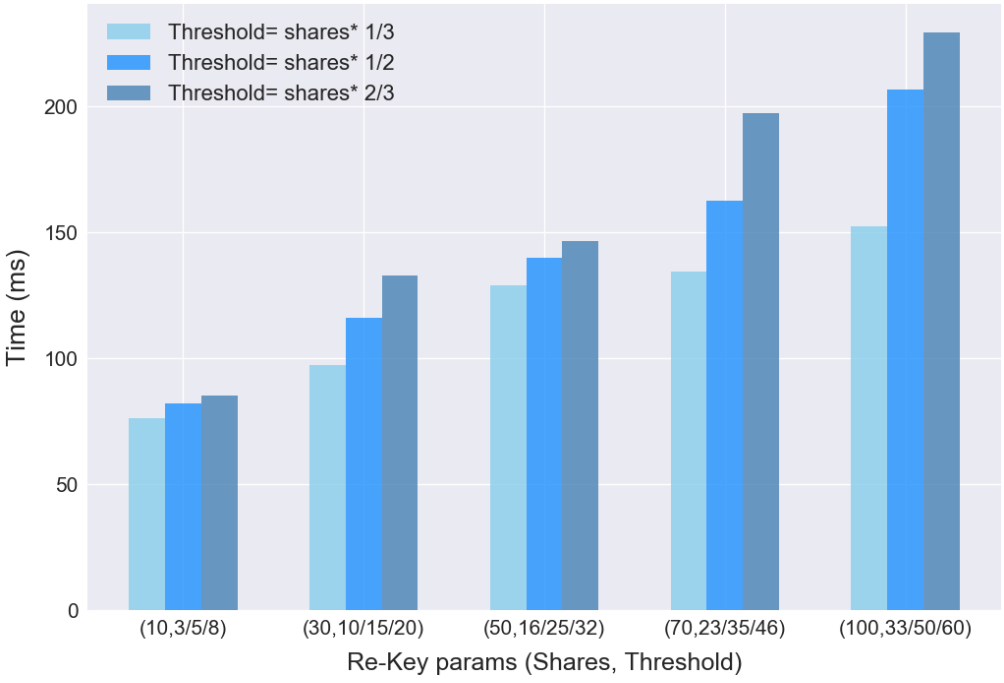

Figure 7. The impact of re-encryption key parameters on data re-encryption processing efficiency.

We measure the scalability of our scheme by performing multiple transactions from multiple data users. In our blockchain simulation experiment, the whole process was repeated 10 times for each scenario before taking the average, and the process was repeated by increasing five requests until the overall request reached 50. The latency is measured from when a data request transaction is submitted to the blockchain network until the data users successfully decrypt the re-encrypted ciphertext. Figure 8 shows a steady and linear increase in the latency, due to the increase in requests. The latency is a result of the

overloads on the flow of transactions in the blockchain networks. The simulation result provides a direction for system optimization and can be used to improve efficiency.

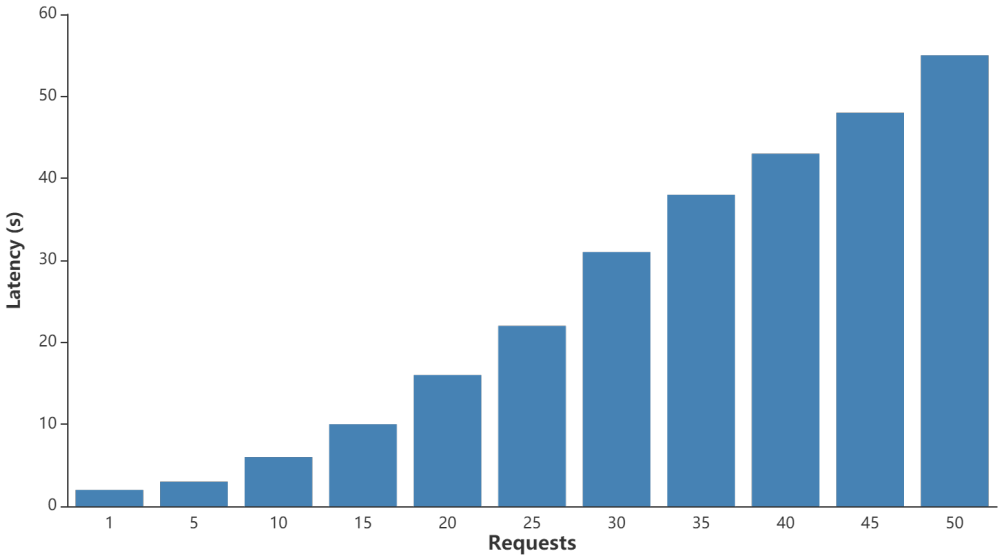

Figure 8. The scalability test.

5.3. System Features and Comparative Performance Analysis

In this section, we mainly compare the performance difference between our scheme and some classic re-encryption mechanism. Table 1 shows the results of functional comparison. Compared with most schemes, our scheme combines proxy re-encryption with the blockchain, so it has the characteristic of decentralization. Moreover, our scheme replaces the single semi-trusted proxy server with multiple consensus nodes in the blockchain network, which reduces the workload of the proxy servers and increases the reliability of the system. In our mechanism, the data owners are only responsible for the generation of re-encryption keys. So the overload of the data owners is also very low. Furthermore, the whole process of data sharing and data re-encryption can be tracked and audited in the blockchain, which can effectively prevent the wrong behavior of malicious nodes.

Table 1. Functional comparison.

	Confidentiality of Data	Decentralization	Multiple Proxies	Split Re-Encryption Key	Do Overload	Proxy Overload	Auditable
[20]	✓	×	×	✓	low	high	×
[21]	✓	×	×	×	high	middle	×
[22]	✓	×	×	×	middle	high	×
[31]	✓	×	×	×	middle	high	×
[32]	✓	✓	×	×	low	high	✓
[33]	✓	×	×	×	high	high	×
[17]	✓	✓	×	×	high	low	✓
Our	✓	✓	✓	✓	low	low	✓

In terms of efficiency comparison, Figure 9 reveals the computation time of each stage of data sharing in various schemes. This figure measured was a result of the average CPU time of 100 executions for each type of operation. In the stage of data encryption and data decryption, compared with other schemes, such as WMXZL [31], SWLX [21], KXECXG [32], our scheme presents slightly less time overload than other schemes. Scheme ZDWQ [33]

has obviously higher time consumption because it is executed for a group of users, using the broadcast encryption method. In the stage of data re-encryption, since threshold re-encryption and the consensus algorithm require higher computational overload, the time latency of our scheme is increased significantly, and it is still much lower than the scheme [33].

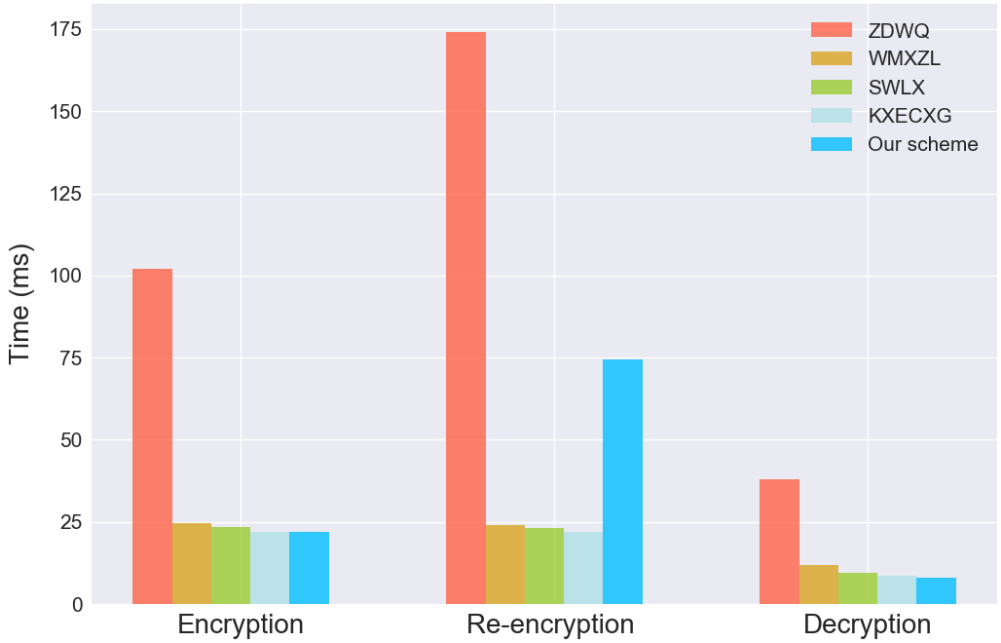

Figure 9. The time-consuming comparison of different stages of data sharing between our scheme and other schemes.

6. Conclusions

In this paper, we propose a new architecture with practical significance for secure IoT data sharing, which allows the data owners to store their encrypted data in the storage service providers and share them with legitimate users efficiently. By converging threshold proxy re-encryption mechanism and blockchain consensus algorithm, our scheme can utilize a decentralized blockchain network to eliminate dependence on the third-party central proxy servers, which supports sharing sensitive data for decentralized and centralized applications. We can realize the re-encryption calculation of encrypted data in the process of consensus confirmation in the blockchain network. Moreover, the records and results of data sharing are permanently recorded in the blockchain distributed ledger, which can ensure the correctness and reliability of data sharing.

We provided a detailed performance and security analysis of the proposed scheme. Simulation experiments revealed that the consensus algorithm in the blockchain can be well combined with threshold proxy re-encryption for data sharing. In our architecture, a single semi-trusted proxy server is replaced by a decentralized consensus node cluster. This is a novel attempt that has not been carried out in other related work. Moreover, our scheme has satisfactory performance and scalability. Compared with other schemes, the time overload introduced while enhancing the security of the IoT data sharing system is acceptable. Our scheme can meet the IoT application scenarios of large-scale access requests. In the future, we plan to expand our solution in the actual deployment scenario further to verify the availability and practicability of the proposed scheme.

Author Contributions: Conceptualization, B.H. and Y.C.; methodology, B.H.; validation, H.Y., Z.D. and J.H.; writing—original draft preparation, B.H.; writing—review and editing, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: The work is supported by the National Key Research and Development Program of China under grant 2018YFB0204301, the National Natural Science Foundation (NSF) under grant 62072306, Open Fund of Science and Technology on Parallel and Distributed Processing Laboratory under grant 6142110200407.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

1. Fu, J.; Liu, Y.; Chao, H.; Bhargava, B.; Zhang, Z. Secure Data Storage and Searching for Industrial IoT by Integrating Fog Computing and Cloud Computing. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4519–4528. [CrossRef]
2. Alladi, T.; Chamola, V.; Sikdar, B.; Choo, K.R. Consumer IoT: Security Vulnerability Case Studies and Solutions. *IEEE Consum. Electron. Mag.* **2020**, *9*, 17–25. [CrossRef]
3. Chen, H.; Yu, J.; Liu, F.; Cai, Z.; Xia, J. Archipelago: A Medical Distributed Storage System for Interconnected Health. *IEEE Internet Comput.* **2020**, *24*, 28–38. [CrossRef]
4. Blaze, M.; Bleumer, G.; Strauss, M. Divertible protocols and atomic proxy cryptography. In Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, 31 May–4 June 1998.
5. Xi, C.; Yun, L.; Yong, L.; Lin, C. Threshold Proxy Re-encryption and Its Application in Blockchain. In Proceedings of the International Conference on Cloud Computing and Security, Haikou, China, 8–10 June 2018.
6. Lou, S.M.; Cao, Z.F. Identity-based proxy re-encryption with threshold multi-proxy. *J. Nat. Sci. Heilongjiang Univ.* **2010**, *27*, 151–156.
7. Huh, S.; Cho, S.; Kim, S. Managing IoT devices using blockchain platform. In Proceedings of the 2017 19th International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea, 19–22 February 2017; pp. 464–467.
8. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Decentralized Bus. Rev.* **2008**, 21260.
9. Mollah, M.B.; Azad, M.; Vasilakos, A. Secure Data Sharing and Searching at the Edge of Cloud-Assisted Internet of Things. *IEEE Cloud Comput.* **2017**, *4*, 34–42. [CrossRef]
10. Wei, W.; Peng, X.; Yang, L.T. Secure Data Collection, Storage and Access in Cloud-Assisted IoT. *IEEE Cloud Comput.* **2018**, *5*, 77–88.
11. Yin, H.; Xiong, Y.; Zhang, J.; Ou, L.; Liao, S.; Qin, Z. A Key-Policy Searchable Attribute-Based Encryption Scheme for Efficient Keyword Search and Fine-Grained Access Control over Encrypted Data. *Electronics* **2019**, *8*, 265. [CrossRef]
12. Luo, W.; Ma, W. Secure and Efficient Data Sharing Scheme Based on Certificateless Hybrid Signcryption for Cloud Storage. *Electronics* **2019**, *8*, 590. [CrossRef]
13. Hui, C.; Xun, Y.; Surya, N. Achieving Scalable Access Control Over Encrypted Data for Edge Computing Networks. *IEEE Access* **2018**, *6*, 30049–30059.
14. Nawaz, A.; Queralta, J.P.; Guan, J.; Awais, M.; Gia, T.N.; Bashir, A.K.; Kan, H.; Westerlund, T. Edge Computing to Secure IoT Data Ownership and Trade with the Ethereum Blockchain. *Sensors* **2020**, *20*, 3965. [CrossRef] [PubMed]
15. Jiang, Y.; Shen, X.; Zheng, S. An Effective Data Sharing Scheme Based on Blockchain in Vehicular Social Networks. *Electronics* **2021**, *10*, 114. [CrossRef]
16. Xia, Q.; Sifah, E.B.; Asamoah, K.O.; Gao, J.; Du, X.; Guizani, M. MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain. *IEEE Access* **2017**, *5*, 14757–14767. [CrossRef]
17. Manzoor, A.; Liyanage, M.; Braeken, A.; Kanhere, S.; Ylianttila, M. Blockchain based Proxy Re-Encryption Scheme for Secure IoT Data Sharing. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Korea, 14–17 May 2019. [CrossRef]
18. Ateniese, G. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.* **2005**, *9*, 1–30. [CrossRef]
19. Ge, C.; Susilo, W.; Wang, J.; Fang, L. Identity-based conditional proxy re-encryption with fine grain policy. *Comput. Stand. Interfaces* **2017**, *52*, 1–9. [CrossRef]
20. Zheng, Z.; Zhang, M.; Dai, X.; Wang, X. Access control with high efficiency scheme for cloud storage based on proxy re-encryption. *Appl. Electron. Tech.* **2016**, *42*, 99–101.
21. Shao, J.; Wei, G.; Yun, L.; Xie, M. Identity-Based Conditional Proxy Re-Encryption. In Proceedings of the IEEE International Conference on Communications, Kyoto, Japan, 5–9 June 2011.
22. Liang, K.; Susilo, W. Searchable Attribute-Based Mechanism With Efficient Data Sharing for Secure Cloud Storage. *IEEE Trans. Inf. Forensics Secur.* **2017**, *10*, 1981–1992. [CrossRef]
23. Mignotte, M. *How to Share a Secret*; Springer: New York, NY, USA, 1982; pp. 371–375.
24. Vukoli, M. The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. In Proceedings of the International Workshop on Open Problems in Network Security, Zurich, Switzerland, 29 October 2016.

25. King, S.; Nadal, S. *PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake*; 2012.
26. Majumdar, M.A.; Monim, M.; Shahriyer, M.M. Blockchain based Land Registry with Delegated Proof of Stake (DPoS) Consensus in Bangladesh. In Proceedings of the 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 5–7 June 2020.
27. Zheng, X.; Feng, W. Research on Practical Byzantine Fault Tolerant Consensus Algorithm Based on Blockchain. *J. Phys. Conf. Ser.* **2021**, *1802*, 032022. [CrossRef]
28. Hu, B.; Chen, Y.; Yu, H.; Meng, L.; Duan, Z. Blockchain Enabled Data Sharing Scheme for Consumer IoT Applications. *IEEE Consum. Electron. Mag.* **2021**. [CrossRef]
29. American National Standards Institute, Inc. (ANSI). *62: Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (Ecdsa)*; American National Standards Institute: New York, NY, USA, 1999.
30. *Announcing the Advanced Encryption Standard (AES)*; Federal Information Processing Standards Publication; National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2001; Volume 29, pp. 2200–2203.
31. Xu, A.W.; Ma, J.; Xhafa, F.; Zhang, M.; Luo, X. Cost-effective secure E-health cloud system using identity based cryptographic techniques. *Future Gener. Comput. Syst.* **2017**, *67*, 242–254.
32. Agyekum, K.O.B.O.; Xia, Q.; Sifah, E.B.; Cobblah, C.N.A.; Xia, H.; Gao, J. A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of Things Based on Blockchain. *IEEE Syst. J.* **2021**, 1–12. [CrossRef]
33. Zhou, Y.; Deng, H.; Wu, Q.; Qin, B.; Liu, J.; Ding, Y. Identity-based proxy re-encryption version 2: Making mobile access easy in cloud. *Future Gener. Comput. Syst.* **2016**, *62*, 128–139. [CrossRef]