Data Sharing and Tracing Scheme Based on Blockchain

Zuan Wang
College of Computer Science and Technology, State Key Laboratory of Public Big Data
Guizhou University
Guiyang, China
vinheres@163.com

Youliang Tian*
State Key Laboratory of Public Big Data, College of Computer Science and Technology
Guizhou University
Guiyang, China
yltian@gzu.edu.cn

Jianming Zhu
School of Information
Central University of Finance and Economics
Beijing, China
zjm@cufe.edu.cn

***Abstract*—The existing data sharing models have some issues such as poor transparency of data transactions, data without security assurance and lacking of effective data tracking methods. This paper proposed a brand-new data sharing scheme based on blockchain technology. Firstly, a blockchain double-chain structure about blockchain was introduced, one chain was used to store the original data and another was used to store transaction data generated by transactions. This structure separated the original data storage and data transactions. Secondly, combined with proxy re-encryption technology, safe and reliable data sharing were achieved. Finally, a new design was implemented. The logical structure of data transaction records enables data to be traced. The results of correctness and security analysis showed that this scheme can provide new technical ideas and methods for big data sharing and data trace.**

Keywords—data sharing, data traceability, blockchain, double-chain structure, proxy re-encryption formatting

I. INTRODUCTION

Data is the third most basic strategic resource after material and energy. At present, the development of the big data industry is facing the dilemma called "data island". Everyone has data, while everyone lacks data. The effective solution is to establish a reasonable and efficient data-shared model. Data open and data transaction are two common data-shared models [1]. Data open originated from the 'data open' movement in the early 21st century. This model advocates using data freely. It includes four stages: the selection of data sets, the development of open protocols or rules, making data accessible and easy to be discovered. Data transaction originated from the stock market or newspaper industry in the middle of the 19th century. It is a kind of data commodity trading behavior based on the value of data [1]. Data transaction is generally implemented by data trading platforms. One of the most representative is the Guiyang big data exchange. It requires data demanders to pay a certain fee before getting data. After the transaction, data providers are paid a certain percentage of the cost by the exchange.

At present, there are few research achievements about data sharing. Lvan et al. proposed a method for securely storing medical records based on blockchain [2]. Zyskind et al. stored the encrypted data off the chain in the form of DHT, and saved the data store address in the public ledger [3]. There are two types of transaction: Taccess and Tdata in the system. The former is used for the authorized accessing and update between a user and a service provider; the latter is used to query and store data. Shrie et al. proposed the use of MIT's OPAL/Enigma encryption platform and blockchain technology to create a secure environment for storing and analyzing medical data [4]. Kuo et al. used a private chain network based on blockchain technology to create an inter-agency medical health prediction model [5].

Although the traditional data sharing model can solve the data using dilemma, there are still problems be solved: data are converged and stored in the single third-party platforms, and data analysis or data usage is out of control from the data providers, so it cannot effectively meet the demands like transparency, trust, control and value [6]. It's the major problem in existing data sharing models. In [3], the authors make good use of blockchain technology to collect, store, and share sensitive data, but don't optimized the data tracing model from the underlying structure of the blockchain. In view of these existing problems, based on blockchain technology [7]-[9] and proxy re-encryption technology [10]-[13], this paper proposed a data sharing scheme based on the blockchain double-chain structure [14]. This schema could ensure the integrity during the data sharing process, the traceability of data sources, the high availability of data sharing, and the prevention of data reselling. During the initialization phase, the data source providers needs to cut the data into some data blocks, in order to facilitate the search or storage of duplicate data. Then the data source providers encrypt the data blocks and store them in the data blockchain (DBC) in the form of cipher text. During the flow and sharing phase, the data source providers generate a re-encryption key for the data users, and broadcast it to the DBC network. Any node can store a re-encryption key. When the data user receives the re-encryption key, the cipher text which was encrypted by the data provider public key can be re-encrypted, and the cipher text encrypted by the data user's public key can be obtained. The data user can use his own private key to decrypt the encrypted cipher text to obtain a shared data block. Then repeat the operation to obtain a complete data block set. At the same time, the trading blockchain (TBC) records this data sharing behavior. During the data traceability phase, using the traceability of the blockchain, the data transaction is signed to implement the query of the data flow path. This paper analyzed the details of data transactions, designed the structure of data transaction and original data storage, and built a new model for data sharing, in order to implement the traceability of

Supported by National Natural Science Foundation of China (Grant No. 61662009, 61772008, U1509214), Ministry of Education - China Mobile Research Fund Project (Grant No. MCM20170401), Science and Technology Major Support Program of Guizhou Province (Grant No. 20183001) and Guizhou University Cultivation Project (Grant No. [2017]5788).

978-1-5386-6965-5/18/$31.00 ©2018 IEEE

data sharing and data flow path. This ensures the data copyright ownership of data source suppliers and provides technical support for the healthy and orderly development of the big data industry.

The main contributions of this paper are as follows: 1) The double-chain structure model of blockchain was designed, and the original data and transactions data were stored separately. In this model, the original data was stored in the DBC with encrypted, and the records generated by data transaction were stored in TBC. Two blockchains used the classical PBFT protocol [15] to confirm and exchange relevant information, which improved the concurrency of the system. 2) A data transaction structure was constructed. In this structure, the data owner signed a random hash digital signature by the previous transaction, data block number, permission level and public key of the next data user. This is conducive to query of data flow path. 3) Combined with the proxy re-encryption technology [16], the security sharing of data was achieved.

The rest of paper is organized as follows. Section 2 proposes a data sharing and tracing scheme based on blockchain. The correct analysis, security analysis and complexity analysis of this scheme were introduced in Section 3. Finally, the conclusions are presented in the Section 4.

II. Data Sharing and Tracing Scheme

A. System Model

The data sharing and traceability model adopts the structure of two new kinds of blockchain as shown in Fig. 1. One is a data blockchain(DBC), which mainly stores original data, and the other is a trading blockchain(TBC), which mainly stores information which is useful for trading and executes transaction. At the same time, both blockchains conform to the following blockchain definition.

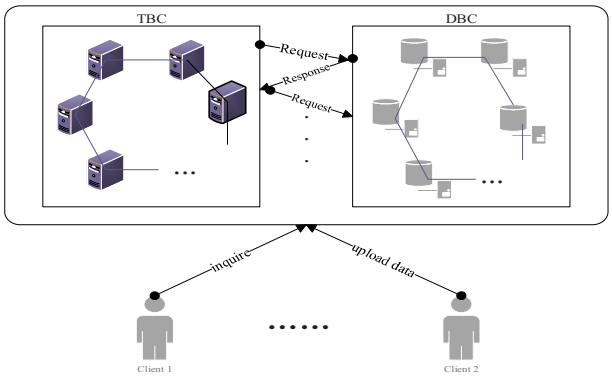

Fig. 1. Structure of two new kinds of blockchain.

- Each block has a timestamp. It has hashed the information of the previous block and each transaction is verified.
- The two blockchains operate independently. At the same time, they also confirm and exchange information about the original data. In order to ensure the authenticity of the information, the model adopts the classic PBFT protocol.
- They use the Byzantine General Question to vote and tolerate less than one-third of nodes cheating or being hacked.

According to the current status of big data transactions, we set users into different types: complete, lightweight, and general users. Lightweight users will use websites or mobile apps to check their ownership data and may authorize or revoke data access. We divide the clients used by users into 3 types.

- Full-featured client: they store all blockchain data (large data providers can provide interfaces for external services). These clients mainly refer to the nodes which generate blocks in the TBC.
- Lightweight client: They does not save the complete blockchain data, They need to query other full-featured nodes which can provide query interface and provide data owners to complete such services as authorization. These clients mainly refer to nodes in the TBC that only have the function of the verification block.
- Online client: They can help people check data and their status through a browser or mobile app. For example, when a user agrees to upload their data by a mobile application, the development company of the mobile application will report to TBC and DBC through a full-featured client. When users want to check their own data, they can obtain their own historical records through authorization queries by lightweight clients. The online client is a simple self-checking service for users.

According to industry background research, companies or organizations that accumulate big data are basically concentrated in large companies. On the one hand, small-scale enterprises have small data and generate less user data; on the other hand, data center construction is not perfect and there is no strong data processing capability. Therefore, we designed a DBC server cluster and a TBC server cluster. Using the improved PBFT consensus algorithm [17], we can effectively use blockchain technology to achieve decentralized, secure and fast data sharing and detection of data flow path.

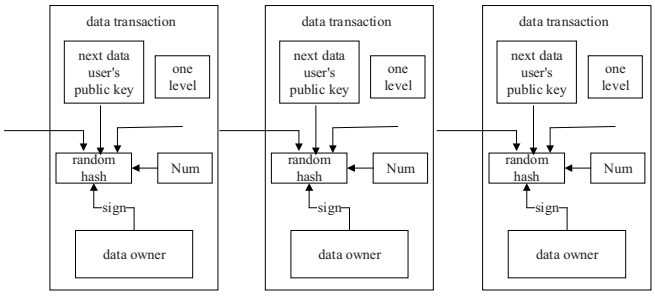

Fig. 2. Data transaction structure.

In order to ensure that the content of data transaction records is credible, tampered and traceable, this paper views the flow of data as a kind of transaction. First, the data needs to be divided into n data blocks. In the transaction, the data owner needs to sign a random hash digital signature on the previous transaction, the data block number, the permission level, and the public key of the next data user and appends this signature after the hash value of this data block. Then the

data is submitted to the next data user. It should be noted that the level of authority granted by the data owner to the next is generally divided into three levels: level one, which has the ability to use data, grant data use rights and grant distribution data use rights; level two, with the ability to use data, grant data usage rights; level three, with the ability to use data. As shown in Fig. 2, it is a structure diagram of a data transaction record.

Nodes in the TBC collect and record data transactions. These transaction data generate a summary and store it in a hierarchical structure. The hash value of the segmented data block is stored in the Item structure, then n such Item structures are calculated, and they are stored in the Item block structure. As shown in Fig. 3, the blocks in the blockchain are composed of a plurality of Item blocks. After calculating the hash value one level after the other, we will get the Merkle root of these Item blocks. Each Item block stores n Items and a header information (as shown in Fig. 4). The header information includes the number of items, the address corresponding to the data stored in the DBC, the Merkle root of the n items, etc., which is not only beneficial to the propagation of each block in a peer-to-peer network, but also reduces the cost and search space for data verification, and speeds up the user verification of records. In general, each Item contains three pieces of information: the data owner's public key, metadata (including data block number), data digest (including signature).

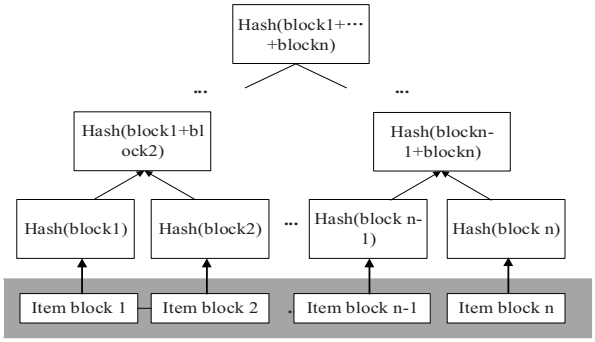

Fig. 3. Merkle tree to store item blocks.

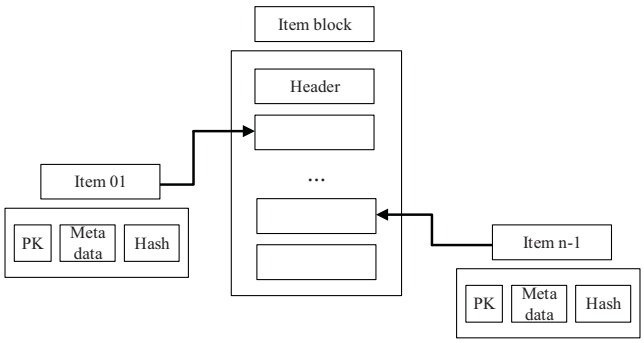

Fig. 4. Data storage hierarchy.

For the nodes in the DBC , they will record the original data. The node will also divide the original data into n data blocks and encrypt them. One data block constitutes an Item in the DBC, and the n items constitute an Item block. The structure of the Item block is similar to that of the TBC, which contains n Items and a header. The header information includes the number of items, the Merkle roots of n items, and the index structure of an Item. This not only facilitates the propagation of each block in a peer-to-peer network, but also speeds up the validation of data by users. Meanwhile, Smaller data blocks prevent data reselling. In general, each Item also needs to contain three pieces of information: the data owner's public key, the metadata (including the data digest), and the encrypted data.

The classical PBFT protocol is widely used in distributed systems. It is a C/S response mode and cannot be used as a peer-to-peer network in the blockchain. The three-phase protocol requires three full-network broadcasts, which will increase the consumption of network bandwidth. Because the PBFT protocol is oriented to a static network structure, it cannot dynamically sense the joining or leaving of nodes. To sum up, we have adopted the improved PBFT protocol in [17], which divides the entire network node into different groups and uses the computational power to expand the throughput to generate new blocks.

B. Scheme Construction

The construction of this paper includes the following three parts: initialization; data flow and sharing; data trace.

1) Initialization

Let data source provider set be $\{P_1,P_2,...,P_n\}$, data user set be $\{U_1,U_2,U_3\}$ and data source supplier server group constitute data blockchain (DBC) and transaction blockchain (TBC). Among them, for a piece of raw data, a server node may be a data source provider. And for another raw data, it is a data user. At the same time, a bilinear map (q,g,G_1,G_T,e) is initialized. $H:\{0,1\}^* \rightarrow \{0,1\}^k$ is a cryptographically unidirectional, collision-resistant hash function using SHA - 256 instantiation. Let g_1 be the generator different from g in G_1 and obtain the public parameter $Param=(q,g,G_1,G_T,e,g_1,H)$.

The data source supplier divides the data D into n data blocks $d_1,d_2,\cdots,d_n \in Z_p$, where p is a large prime number and $D=\{d_z\}(z\in[1,n])$. Data blocks are the basic unit for data sharing and tracing. A data source supplier P_i selects a random number x in Z_p^* and generates a key pair (pk_i, sk_i), where $pk=g^x$ and the random number x is the private key sk_i.

The data source provider P_i uses the public parameters $param$, public key pk, and data plaintext d_z to obtain the first encrypted cipher text. The encryption process is as follows: a random number r_i is selected in Z_p^* and a re-encrypted cipher text C_z is calculated by (1)-(4).

$$C_z=(c_1,c_2,c_3) \tag{1}$$

$$c_1=g^{r_i} \tag{2}$$

$$c_2=d_z\cdot e(g_1,pk_i)^{r_i} \tag{3}$$

$$c_3=H(H(c_1)\,\|\,H(c_1\,\|\,c_2)) \tag{4}$$

Then output the cipher text set is $\{C_z\}(z \in [1,n])$ .The cipher text set is stored in the DBC through an improved PBFT consensus algorithm. We also supplement the decryption operation on the data cipher text C_{iz}. Firstly, We first verify whether (5) holds.

$$c_3 = H(H(c_1) \| H(c_1 \| c_2)) \tag{5}$$

If not, return $\perp$. Otherwise, the plaintext is calculated by (6).

$$d_z = c_2 / e(c_1, g_1^{sk_i}) \tag{6}$$

2) Data Transfer and Sharing

When the data source provider P_i wants to share the data D to a certain data user U_j, the P_i needs to generate a proxy re-encryption key based on the public key pk_i, the private key sk_i and the public parameter *param*. By using the generated proxy re-encryption key, the data user U_j can convert the cipher text of pk_i into the cipher text pk_j. Thus, U_j can use its own private key to decrypt the cipher text originally encrypted by pk_i. The process of the re-encrypting key is as follows. Firstly, we should select a random number r_j in Z_p^* and calculate the equations.

$$rk_{i \to j} = (rk_1, rk_2, rk_3) \tag{7}$$

$$rk_1 = g^{r_j} \tag{8}$$

$$rk_2 = g_1^{-sk_i} pk_j^{r_j} \tag{9}$$

$$rk_3 = H(H(rk_1) \| H(rk_1 \| rk_2)) \tag{10}$$

Then the re-encryption key $rk_{i \to j}$ is output.

When the data source provider P_i broadcasts $rk_{i \to j}$, the data sharing or delivery process is completed. Because U_j obtains $rk_{i \to j}$, it can complete the transformation of the cipher text. The specific operations are as follows:

- Verify (11) and (12).

$$c_3 = H(H(c_1) \| H(c_1 \| c_2)) \tag{11}$$

$$rk_3 = H(H(rk_1) \| H(rk_1 \| rk_2)) \tag{12}$$

If one of the above equations does not hold, it shows that there is a problem with $rk_{i \to j}$, and it needs to resend. Otherwise, continue 2) operation.

- Calculate the cipher text set $\{W_j\}(j \in [1,n])$ encrypted by the U_j public key pk_j to convert the cipher text.

$$W_j = (W_1', W_2', W_3', W_4') \tag{13}$$

$$W_1' = c_1, W_2' = c_2 \cdot e(c_1, rk_2), W_3' = rk_1 \tag{14}$$

$$W_4' = H(H(W_1') \| H(W_1' \| W_2') \| H(W_2' \| W_3')) \tag{15}$$

If U_j wants to obtain the plaintext of the data, he only needs to decrypt it with its own private key. The specific steps are as follows:

Firstly, We verify whether (16) holds.

$$W_4' = H(H(W_1') \| H(W_1' \| W_2') \| H(W_2' \| W_3')) \tag{16}$$

If not, return $\perp$. Otherwise, calculate and output the plaintext of the data block d_z in sequence.

$$d_z = W_2' / e(W_1', W_3')^{sk_i} \tag{17}$$

Similarly, the data $D = \{d_z\}(z \in [1,n])$ can be obtained, and data sharing or transaction can be implemented.

After the data sharing or transaction has been completed on the DBC, n data transaction records (because the data is divided into n copies) will be formed on the TBC and the metadata will be recorded. The nodes on the TBC query the authenticity of the transaction through the classic BPFT protocol. When true, the transaction is placed in the transaction pool. Otherwise, it is discarded.

3) Data Trace

In the process of data sharing, data copyright infringement is very likely to occur. The blockchain node verifies each transaction and the blockchain also has very good traceability, which will prevent the reselling behavior of the data. At the same time, it can provide users with query of data flow path.

When the data source provider P_h divides the original data D into n data blocks, some of which have fraudulent behavior and assume that one of the data blocks d_k is misappropriation data. When this data block is packaged as Item is spread on the TBC network. The node that receives the Item will check against a list of standard verification processes. The standard generally includes:

- Item format fill is valid.
- Metadata information is complete.
- Using classic PBFT protocol to interact with DBC to determine whether data block d_k satisfies integrity.
- Through the index structure (in the Item block header information in DBC) we can check (18). h is a data block that already exists in the data blockchain. If (18) holds, you can use the signature to judge whether it is shared data. If it is not shared data, then it is the behavior of misappropriation of data. Otherwise, it means that this is a raw data block.

$$H(d_k) = H(h) \tag{18}$$

When any one of them is not satisfied, the node will discard the transaction. Especially when the node finds that

the fourth point is not satisfied, the node will broadcast the misapplied data block on the entire network, which will cause the data source provider P_h to submit the original data failed.

If d_k is a shared data block, we can find the signature from P_s and metadata related information. And we can complete the query path from P_h to P_s. Similarly, we can continue to trace the source of the original data block based on the signature stored in the Item of P_s.

III. Case Analysis

A. Correctness Analysis

Theorem 1. Firstly verifies the correctness of (6) and (17).

Proof: According to the above, $\{C_z\}(z \in [1,n])$ is the cipher text set and the cipher text of the data block is $C_z = (c_1, c_2, c_3)$. Because of the bilinearity of the bilinear map, the following equation can be obtained.

$$\begin{aligned} d_z &= \frac{c_2}{e(c_1, g_1^{sk_i})} \\ &= \frac{d_z \cdot e(g_1, pk_i)^{r_i}}{e(g^{r_i}, g_1^{sk_i})} \\ &= \frac{d_z \cdot e(g_1, g^{sk_i})^{r_i}}{e(g^{r_i}, g_1^{sk_i})} \\ &= \frac{d_z \cdot e(g_1, g)^{sk_i r_i}}{e(g, g_1)} = d_z \end{aligned}$$

Equation (6) can be proved.

Because the shared or transaction data cipher text set is $\{W_j\}(j \in [1,n])$ and the ciphertext of the data block is (13), (14) and(15). According to the bilinearity of the bilinear map, the following equation can be obtained.

$$\begin{aligned} d_z &= \frac{W_2^{'}}{e(W_1^{'}, W_3^{'})^{sk_i}} \\ &= \frac{c_2 \cdot e(c_1, rk_2)}{e(c_1, rk_1)^{sk_i}} \\ &= \frac{d_z \cdot e(g_1, pk_i)^{r_i} \cdot e(g^{r_i}, g_1^{-sk_i} pk_j^{r_j})}{e(g^{r_i}, g_1^{r_j})^{sk_i}} \\ &= \frac{d_z \cdot e(g_1, g)^{sk_i r_i} \cdot e(g^{r_i}, g^{-sk_i} (g^{sk_j})^{r_j})}{e(g^{r_i}, g^{r_j})^{sk_i}} \\ &= \frac{d_z \cdot e(g_1, g)^{sk_i r_i} \cdot e(g, g_1)^{-r_i sk_i} \cdot e(g, g)^{r_i r_j sk_i}}{e(g, g)^{r_i r_j sk_i}} = d_z \end{aligned}$$

Equation (17) can be proved.

B. Security Analysis

This section will analyze data sharing and tracing schemes from two aspects of data integrity and traceability.

Theorem 1. This data sharing and tracing scheme satisfies data integrity.

Proof: On the one hand, it refers to the integrity of data D. Item block has the total number of Item in the DBC and each Item contains the data block number, so that the data block in the data D can be guaranteed not to be lost. On the other hand, it refers to the integrity of the data block d_z during the transaction or sharing of data. Since the data source provider P firstly divide the data D into $d_1, d_2, \ldots, d_n$. In the DBC, the data block d_z is encrypted to obtain the cipher text c_z. If the data user U_i obtains the data block d_z shared by the data source provider P, he falsifies and modifies some of the data in the d_z. Then he re-encrypts it to get the data cipher text w_z and attempts to submit it to the blockchain, so he needs to broadcast w_z to each node in the DBC. When the consensus node in the DBC receives the Item, it re-encrypts c_z by using the broadcast re-encryption key to obtain $W_z^{'}$ and checks (19).

$$H(W_z^{'}) = H(W_z) \tag{19}$$

Unless the data user U_i can find a $W_z^{'} \neq W_z$ make $H(W_z^{'}) = H(W_z)$, it is obviously impossible to be equal. Thereby, this can ensure the integrity of the data block d_z.

Theorem 2. This data sharing and tracing scheme satisfies data traceability.

Proof: When the data source provider P_h divides the original data D into n data blocks $d_1, d_2, \ldots, d_n$, and the data block d_k is traceable data. When the data block is packaged as Item to propagate it on the TBC network, consensus node will check the Item and extract $H(d_k)$, the signature s, and the data owner P_g's public key pk_g from the Item transaction data. The signature can be verified by pk_g. If it is successful, the previous transaction, data block number, permission level and the public key of the next data user $pk_h^{'}$ (the public key of P_h) can be restored. The restored $pk_h^{'}$ is not the public key of the data source provider P_h, which proves that P_h hijacked the data block d_k. On the contrary, we can know that the data transaction is legal, and the data block is traded by P_g or shared to P_h. Similarly, we can further check the Item submitted by P_g to further confirm the source of d_k.

C. Complexity Analysis

As shown in Table 1, it is a comparison of the functions of scheme in literature [3] and this paper. However, literature [3] does not give complexity analysis, and no comparative analysis is done here. The complexity analysis of this scheme in three stages of data storage, data sharing and tracing is given below. The communication complexity of this solution is represented by the number of communication rounds. Table 2 lists the complexity analysis of this scheme.

The computational complexity of the data storage stage is $o(n)$ and the communication complexity is $o(nM(N-l))$, where n is the number of data blocks, N is the number of consensus nodes in the DBC, l is the number of Byzantine nodes in the

DBC and *M* is the number of consensus nodes in TBC. The nodes in the TBC want to use the improved BPFT consensus for consistency confirmation and *M* nodes in the TBC need to perform *(N-l)* round confirmation for n data blocks, so the communication complexity is *o(nM(N-l))*. The computational complexity of the data sharing phase is *o(1)* and the communication complexity is also *o(1)*. The calculation complexity of the data tracing is *o(k)* and the communication complexity is *o(1)*, where k is the number of data flow nodes passing through.

TABLE I. COMPARISON OF THE FUNCTIONS

	functions of scheme	
	literature [3]	***This paper***
Data collecting	√	×
Data storage	√	√
Data sharing	√	√
Data Tracing	×	√

TABLE II. COMPLEXITY ANALYSIS

	Complexity	
	Computational complexity	***Communication complexity***
Data storage	$o(n)$	$o(nM(N-l))$
Data sharing	$o(1)$	$o(1)$
Data Tracing	$o(k)$	$o(1)$

IV. CONCLUSION

This paper introduced the double-chain structure of the blockchain to separate the data transaction records from the original data. This obviously improved the throughput of the blockchain and reduced the communication pressure of the single-chain blockchain. Encryption technology was used to store the original data to ensure data privacy and proxy re-encryption technology was used to achieved data sharing on the blockchain. In summary, the solution satisfied the needs of data privacy protection and data sharing with large traffic and fast response. In the design of this scheme, there are still some deficiencies, such as the inability to detect the source of data leaks, the lack of theoretical support for the segmentation of data blocks and etc. These will be the direction of our next research and improvement, and help find someone who can claim responsibility for data leakage among the data transaction life.

ACKNOWLEDGMENT

Firstly, I would like to show my deepest gratitude to Youliang Tian, a professor of Guizhou University , a respectable, responsible and resourceful scholar, who has provided me with valuable guidance in every stage of the writing of this thesis. Without his enlightening instruction, impressive kindness and patience, I could not have completed my thesis. His keen and vigorous academic observation enlightens me not only in this thesis but also in my future study. In addition, I would also like to thank Professor Jianming Zhu for his valuable suggestions. Last but not least, I'd like to thank all my friends, for their encouragement and support.

REFERENCES

[1] X. Dong, B. Guo, Y. Shen, X. Duan, Y. Shen, and H. Zhang, "An efficient and secure decentralizing data sharing model," Chinese Journal of Computer, Vol. 41, No. 20, 2018.

[2] "Moving toward a blockchain-based method for the secure storage of patient records," *healthit.gov*, 2016. [Online]. Available: https://www.healthit.gov/sites/default/files/9-16-drew_ivan_20160804_blockchain_for_healthcare final.pdf.

[3] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: using blockchain to protect personal data," IEEE Security and Privacy Workshops IEEE Computer Society, vol. 36, no. 29, pp. 180-184, 2015.

[4] "Blockchain and health IT: algorithms privacy and data," *White paper* , 2016. [Online]. Available: https://www.healthit.gov/sites/default/files/1-78-blockchainandhealthitalgorithmsprivacydata_whitepaper.pdf.

[5] "ModelChain: decentralized Privacy-Preserving healthcare predictive modeling framework on private blockchain networks," *arxiv.org*, 2018. [Online]. Available: https://www.healthit.gov/sites/default/files/10-30-ucsd-dbmi-onc-blockchain-challenge.pdf.

[6] K. Schwab, A. Marcus, J. O. Oyola and W. Hoffman, "Personal data: the emergence of a new asset class," Geneva: Forum World Economic, Technical Report, 2011.

[7] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," IEEE Transactions on Dependable & Secure Computing, unpublished, pp. 1-1, 2016.

[8] S. Huckle, R. Bhattacharya, M. White, and N. Beloff, "Internet of things, blockchain and shared economy applications ☆," Procedia Computer Science, vol. 98 , no. 67 , pp. 461-466, 2016.

[9] Y. Yuan, and F. Y. Wang, "The development status and prospects of blockchain", Acta Automatica Sinica, vol. 42, no. 4, pp. 481-494, 2016.

[10] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," Lecture Notes in Computer Science, vol. 1403, no. 10, pp. 127-144, 1998.

[11] B. Libert and D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy re-encryption," IEEE Transactions on Information Theory, vol. 57, no. 3, pp. 1786-1802, 2011.

[12] Y. Aono, X. Boyen, T. P. Le and L. Wang, "Key-Private proxy re-encryption under LWE," Progress in Cryptology – INDOCRYPT 2013. Springer International Publishing, vol. 8250, no. 1, pp. 1-18, 2013.

[13] H. Wang, Z. Cao and L. Wang, "Multi-use and unidirectional identity-based proxy re-encryption schemes," Information Sciences,vol. 180, no. 20, pp. 4042-4059, 2010.

[14] W. T. Tsai, R. Blower, Y. Zhu and L. Yu, "A system view of financial blockchains," 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE), vol. 7, no. 71, pp. 450-457, 2016.

[15] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," ACM Transactions on Computer Systems, vol. 20, no. 4, pp. 398-461, 2002.

[16] G. Ateniese, K. Fu, M. Green and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," Acm Transactions on Information & System Security, vol. 9, no. 1, pp. 1-30, 2006.

[17] L. Luu, V. Narayanan, K. Baweja, C. Zheng, S. Gilbert and P. Saxena, "SCP: a computationally-scalable Byzantine consensus protocol for blockchains," Cryptology ePrint Archive, Report 2015/1168, 2015.