A Private Data Protection Scheme Based on Blockchain under Pipeline Model

Qianyi Dai*, Kaiyong Xv, Song Guo, Leyu Dai, Zhicheng Zhou
Zhengzhou Information Science and Technology Institute, 450001, Zhengzhou, China
*Correspondence author: qianyi.dai@outlook.com

***Abstract*—In order to solve the security problem that private data can be easily disclosed in the centralized storage mode of the Trusted Third Party (TTP), and to realize the dynamic trusted security storage of private data in distributed environment, we designed a distributed user private data protection scheme based on blockchain technology. In the scheme, secondary sharding algorithm is used to encrypt individual private data and the blockchain storage model under the designed channel system improves the sharding data storage scheme and solves the problems caused by the private data centralization storage. Through the theoretical analysis and experiment of the scheme, the designed scheme of this paper has high security, data synthesis efficiency and accuracy.**

I. Introduction

In the era of big data, the amount of data on the Internet is increasing rapidly, and all kinds of data are constantly emerging, collecting and forecasting. Human beings develop from a data-driven society, but they also face the problems of data control. A large number of personal and sensitive data (such as personal medical data, asset inventory, etc.) are stored in a centralized storage organization, which is often not expected to be disclosed, but the centralized data storage organization prone to a single point of failure, internal and external attacks, which puts users under the risk of unknown user private data being stolen, and poses serious data privacy threat to the user. In recent years, the loss of private data reported by public media repeatedly [1] illustrates the vulnerability of the existing centralized privacy protection system. A distributed solution is urgently needed to solve the problem of high security risk of TTP centralization storage data.

The blockchain was originally proposed by Satoshi Nakamoto [2] as an untamable and traceable blockchain data structure through transparent and trusted rules in an open Peer to Peer (P2P) environment that implements data sharing, auditing, and management in a network of multiple sites or organizations. The blockchain provides a centralized, trusted, distributed TTP data storage platform that can meet the integrity of user private data, non-tampering and auditable security storage needs. Therefore, in this paper, the blockchain is used as a means of personal privacy protection, and the private data secondary sharding protection algorithm is proposed. Also in this paper, the blockchain storage model based on pipeline system is designed for private data storage, which has high security.

978-1-5386-4870-4/18/$31.00 ©2018 IEEE

A. Related Work

Scholars at home and abroad have little research in the field of blockchain technology. Especially, the research on the application of blockchain technology to the protection of personal private data is very limited. Based on the theory of blockchain technology for personal privacy protection, literature [4] takes personal health privacy as the background of third party organization and analyzes the rationality of blockchain technology as the security mechanism to protect the private data of personal health. However, literature [4] does not give the technical details of the application scenario implementation of blockchain protection for individual privacy protection.

For technical details of using blockchain technology to solve the private data security problem, Guy Zyskind et al. [5] proposed a decentralized personal information management system. The system uses the traditional bitcoin system to transmit instructions such as storage, inquiry and data analysis, so as to ensure the security of users' privacy information; however, there is no specific case application scenario. Ahmed Kosba et al. [6] proposed the Hawk decentralized intelligent contract system that stores the privacy of the transaction through ciphertext in the blockchain in order to solve the problem of the lack of privacy protection mechanism of transaction data in the process of blockchain trading; however, this method can only protect the privacy of transaction information and cannot protect the security of user's personal identity data. Amir Lazarovich [7] used the blockchain technology to protect privacy by proposing the third-party database escrow based on distributed storage and the audit system based on blockchain and took medical information personal privacy protection as an example to illustrate the application of blockchain technology in personal privacy.

B. Contribution of the Paper

1) In this paper, a blockchain data storage model based on pipeline system is designed, the private data secondary sharding protection algorithm is proposed, and under the blockchain storage model based on pipeline system, private data distributed storage is conducted to realize the security protection and storage of private data in dynamic and insecure network environment. 2) This paper discusses the anti-aggression and security of the scheme in the complex and unsafe network environment. 3) The experimental results show

that the scheme has high data synthesis efficiency and data synthesis accuracy.

C. Structure of the Paper

Section II discusses the reality environment problems and presents corresponding principles and ideas. Section III gives the design framework of this article and puts forward the block process based on pipeline system. Section IV describes the private data secondary chip processing algorithm designed in this paper. Section V conducts security analysis on the design scheme of this paper. Section VI describes the scheme's private data synthesis efficiency test and synthesis accuracy test. Section VII is the summary and conclusion.

II. Problem Analysis

The purpose of this paper is to solve the problem of the illegal theft of private data in the dynamic and insecure network environment. Our focus is on that in the dynamic Internet environment, the malicious node impersonates the benign node to obtain the network data using legal means and at the same time, malicious nodes have the ability to monitor data, deactivate data and delete data. In this context, to minimize the possibility of malicious nodes to obtain personal private data is a significant task. In view of this, we believe that the design of the private data security protection program should follow the following principles:

Private data security principle: We believe that users have the right to own their own data unconditionally. Users can access and recover or delete their data at any time and choose to destroy or redeploy the personal private data storage scheme at any agent service node. Even if there are illegal nodes in the network to eavesdrop on network communication, it is difficult for illegal nodes to restore user's private data through wiretapping data.

Private data auditability principle: The data processing and access methods of nodes in the network should be fully transparent. All nodes evaluate and monitor the private data stored procedure. All nodes are responsible for the data results. The process and result of collecting private data by proxy service nodes are arbitrable and auditable.

III. Proposed Scheme

A. Designed Architecture

Based on the blockchain system, we designed the private data control architecture of the center in the node domain based on the consensus mechanism which adopts the blockchain architecture based on the accounting system [2]. All nodes in the network can read transaction data or submit transaction applications and verify the validity of transaction data. Because the blockchain is in the transaction process, the transaction ledger data is transparent to the whole node, and the transaction accounting legality is verified by all nodes in the network. In order to maintain the consistency of distributed ledger and ensure the credibility of the data recorded on the blockchain, all transaction ledger data in the blockchain must be publicly available to all participating nodes of the network (The blockchain technology supports the light node mode, not storing all the data, but applying the request data to other full nodes when needed). Therefore, based on blockchain ledger technology, it is impossible to protect privacy by centralizing storage. Instead, it will focus on protecting the identity and data path of the transaction. That is, although all transaction details are visible, the attacker cannot find the true identity information and data storage path of both parties based on the transaction data. Thus, secure storage of private data is implemented. We designed the following system framework.

User (U). The user is the node identity subject with corresponding private data and represents the user identity. Users have full ownership of personal data. The user's personal account information includes the user ID, password, and creation timestamp. They all need to be stored correctly in the user node. All users can create user ID. However, because there is no central server in the blockchain network, ID collision cannot be checked. In this paper, we recommend naming the user ID with a unique identifier.

Data Interactive Audit Server (*DIAS*). As a block in the chain of user private data agency service organization, *DIAS* is responsible for the encrypted user private data sharding, distributed storage and data synthesis, and audit and arbitration on the results of all the data operation. *DIAS* ensures that all relevant data operation information is safely and fully recorded in the blockchain and is responsible for the data. We consider *DIAS* as the infrastructure of the blockchain network with initial security.

Blockchain Network (*BN*). In this paper, *BN* structure refers to pure P2P network structure, and the nodes are equal to each other and have no central control node. The node in *BN* has its own node identifier ID, private and private key pair, and the only IP address and port number used for communication. Each BN node can enable or disable the following service functions: • List creation service: create a list of new storage nodes as a new block on the blockchain. • Bulletin board service: collect broadcast information, verify the legality of information, and release it to the blockchain network. • Relay service: forwards the communication between the unspecified *BN* nodes. • Storage service: stores and recovers data in the specified *BN* node. Naturally, U and *DIAS* in *BN* network are nodes in *BN*, and all *BN* nodes form DCO (Distributed Collaboration Organization) for data management.

Storage Node List (*SNL*). *SNL* is a random sort list of the information of the *BN* node for the storage node. The information contained in the list contains the ID, IP address, public key, and signature information of each *BN* node. After the new *SNL* is created, the new *SNL* is encrypted by *DIAS* and then partitioned and encapsulated in the block. Each encapsulated block is recognized by the multi-party authentication process of *BN* and is recorded in the blockchain with integrity.

Due to the untamperability and transparency of the blockchain, complete security sharing without removing or changing is feasible. As a result of random selection, the sequence of records that may result in the selected person nodes in each list can differ from those of other nodes. This

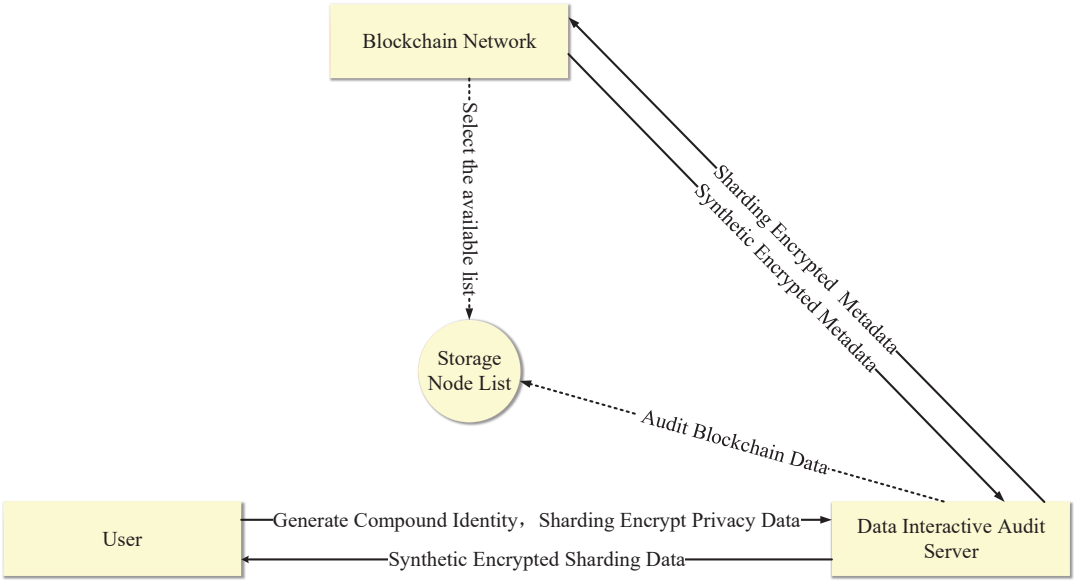

Fig. 1. Personal privacy protection solution based on Blockchain

difference can increase the complexity and security of the combination of storage and restore operation candidate nodes, which can increase the difficulty of the adversary dictionary attack, so we recommend each node to create a new list as different from the existing list as possible. But we use *DIAS* as the final trusted source.

The security mechanism framework of this paper is shown in the figure.

- User U and data interactive audit platform *DIAS* carry out the initial key exchange, and the user grants the data interactive audit platform authority.
- Users upload encrypted privacy information to *DIAS* and *DIAS* fragments data. Implement distributed multi-node storage through data distribution, and the data storage list is submitted to the whole network multiple authentication into the blockchain.
- Illegal users cannot recover and decrypt data without knowing the data storage scheme and the user's private key, so as to ensure the security of users' private data.

B. Pipeline Trading Model Based on Data Interactive Audit Server

In order to solve the problem that in the traditional blockchain mode, the *BN* node sends the block to the network to adopt the flood propagation mode, which leads to the data congestion and the large communication delay. To solve the above problem, we refer to the pipeline system model of the processor and use *DIAS* to connect to U and *BN* and to buffer the transaction data of authentication. *DIAS* makes unified block encapsulation of multiple transaction data of U hidden storage, as the block for the authentication transaction, and improves the efficiency of the block utilization. Then all the *BN* nodes have multiple authentication of contents in the block. Unlike bitcoin, transactions in this paper are not financial and monetary. The transactions referred to in this paper are stored in a fixed format in the block book. Transaction data is primarily the result of formatting the encrypted values of specific data (timestamps, passwords, and identities, etc.). The book refers to the total number of transactions registered by *DIAS* for a period of time. In particular, we give the corresponding block structure of the design, as shown in Table I.

To ensure that the results of all nodes in *BN* are identical, the following procedure is performed by all *BN*. Suppose there are n signed registered nodes in *BN*. $S_m = \{T_{D1}, T_{D2}, \ldots, T_{Dn}\}$ is the account that is sent to *DIAS* and there are many transactions in each transaction book. According to the Paxos consistency algorithm, *DIAS* processes transactions in S in the following process: *DIAS* moves transactions from unordered *Sm* to an ordered list. *DIAS* creates a block for these transactions and puts the block into the blockchain network for all the *BN* nodes to compete for billing, where the result of the billing is the voting value. The legitimacy of the block is determined by the overall voting result. The result is an ordered list of blocks. Each block connects to its parent block and its own data to form the corresponding blockchain.

In the process of multi-party authentication in the blockchain, it should be noted that only signed nodes in *BN* have the right to vote on the validity of a block. All BN nodes determine its validity. There are only two voting results (valid or invalid). Each block is "undecided" when it has not voted on the signed node. When each node receives more than half of the same voting results in the network, the state of the block can be changed from "undecided" to "decided_valid" or "decided_invalid". Once the status result is determined, it cannot be changed. Only valid transactions in the block can

TABLE I
THE BLOCKCHAIN STRUCTURE DESIGNED IN THE PAPER

Blockcontent	Block part	Note	Strlen/bit
ID	Block head	The user ID, which is the Base58 anonymous value of the public key	8
Prev Block Hash	Block head	Hash value of the previous block head	32
TimeStamp	Block head	Timestamp, based on user system generation	4
Version	Block head	Blockchain version number, 1.0 in this paper	4
Merkle Root Hash	Block head	Merkle Root value for transaction content	32
Block Height	Block head	Block height	4
Fulfillments	Block book	Satisfy the list of conditions, the signature value and hash value of each book	72
Operation	Block book	Data manipulation type	2
Data	Block book	Transaction data	-
Data_Hash	Data	Hash value of the previous block head	32
Payload	Data	Data load, related data values	-

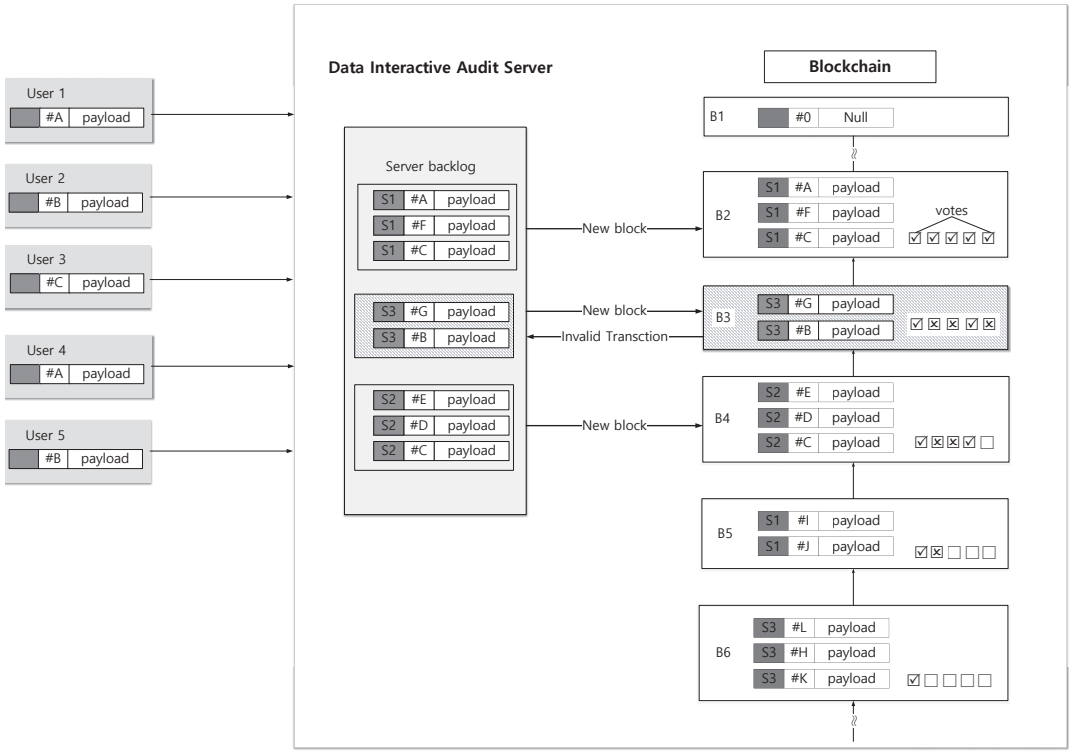

Fig. 2. Multi-Node based on pipeline transaction Blockchain model transaction process.

be written to the blockchain; if there is an invalid transaction in the block, the scheme gives it a chance to re-authenticate it and insert it into the later block to be authenticated. In order to avoid the blockchain bifurcation caused by the state of the parent block, we believe that the transaction of any new block cannot depend on the transaction content in the undetermined block. When multiple votes are cast, any such block will be considered an invalid block. Figure 3 depicts the running process of the blockchain model based on pipeline transactions on the multi-*BN* node.

In the process of the business operation in which the design scheme of this paper is involved, each single block contains various types of transactions of large amounts. In order to reduce the possibility of the adversaries obtaining the transaction content through logic splicing as much as possible, compared with the traditional scheme for block building [10] [11], we believe that under the premise of ensuring the efficiency of block building, special attention should be paid to the randomness of transaction sorting within the block. Therefore, we designed a quick sequencing process of block transactions based on the weight of evaluation of the block transaction. The block body sent by U based on $DIAS$ is considered a decision unit of the transaction set. The number of transactions each block contains is marked as s, and the transactions are denoted as $\boldsymbol{Blockbody} = [Tx_1, Tx_2, , Tx_s]$. This is a typical unit decision process with s inputs and s corresponding outputs. The R-Random transaction sequencing method is adopted. The basic procedures are as follows: a

random number $j(1 \leq j \leq s)$is generated and compared with the previous R number of existing random numbers. If the distance between j and the previously-existing random number in comparison is less than R, then the number j is discarded; Repeat the generation process of the random number j; otherwise, record j, then look for the next random number that fit the premises until all s values are found. We abstract this condition into a Boolean algebra $F(j) \in \{0, 1\}$. The value of j that makes the equation $F(j) = 1$ stand is forcibly set. Then the ensuing search continues.

Step1: Initialize the transaction sequencing array **Block-body**. ***Blockbody*** $[i]$ = $Tx_i, i \in [1,s]$,t=s;

Step2: Generate a random number $a_{rand} \in [1,t], q = a_{rand}$;

Step3: **If** $(F[a_{rand}] = 1)$, jump to step6;

Step4:$a_{rand} = a_{rand} mod$t+1;

Step5: **If**$(a_{rand} > q)$,jump to step3;

Step6: Interchange ***Blockbody*** [arand] and ***Blockbody*** $[t]$, t=$t - 1$;

Step7:**IF**$(t > 1)$,jump to step 2.

Or else, output the transaction sequencing array ***Blockbody*** .

Thus, E_j is the sequence of the transaction in the block-body, and the transactions are renumbered according to the value of E_j, thus forming the block transaction body ***Blockbody***=$[Tx_{E1}, Tx_{E2}, , Tx_{Es}]$. From this, we can get a random sequence of block transactions, making it difficult for adversaries to obtain complete transaction information through exhaustive splicing of all the transactions without knowing the exact location of the transaction, thus ensuring the security of transaction contents.

IV. Private data Secondary Sharding Algorithm

In this section, we describe the design of the private data secondary sharding algorithm, and the data stored procedure is carried out in the *DIAS* blockchain transaction pipeline model. The operation process includes the initialization of the node, the storage operation of the user's private data, and the data synthesis and decryption. In the structure of this paper, we assume that the *BN* is composed of enough network nodes, including a large number of U and *DIAS*, and each node storage system maintains stable state. In particular, we assume that the nodes responsible for data storage are stable online.

Because *BN* does not have a central service node, the data information between the *BN* nodes is communicated through the Gnutella protocol, and nodes periodically broadcast node ID, IP, and port numbers. The node gives *DIAS* the private data protection service through the broadcast public key and the signature of its enabling service function and broadcasts the notification to other nodes for the security of the application service node. If the node detects a malicious application service node, the node also broadcasts information about the malicious application service node and proposes changes to the node storage list. Based on the traditional P2P network bulletin board service, nodes in *BN* can archive the broadcast information of legitimate nodes and malicious nodes and broadcast them continuously and offline nodes can obtain unreceived information broadcast during offline time through the bulletin board service.

In order to illustrate the operation process more clearly, we do the following algorithm description in Table–II.

A. Initialization Operation

We adopt the block designed in table 1, and *BN* adopts the traditional P2P network protocol system. The symmetric encryption algorithm is defined as (Gsym, Esym, Dsym) which are symmetric key generator, symmetric encryption and decryption algorithm; the elliptic curve encryption algorithm consists of three elements (Gecc, Eecc, Decc), which are encryption key generator, encryption algorithm and decryption algorithm; the signature algorithm consists of three elements (Gsig, Ssig, Vsig), which are signature key generator, signature algorithm and verification algorithm. The secp256k1 curve of ECDSA is selected based on the encryption and signature algorithm, and the hash function H adopts the SHA-256 algorithm. (Gpassword, Grandom) is the password generator and the random generator. The password and random number generated by Gpassword and Grandom are fully random. In the blockchain, U uses the compound identity mechanism for identity information protection, and when the key exchange between U and *DIAS* is exchanged, the identity information is shared through protocol 1. And for the identity generation of U and different *DIAS*, different IDs and keys can be used. The composite identity information that is exposed in the network is two-tuple: $Compound^{public}_{U,DIAS}$ = ($pk^{U,DIAS}_{sig}$, $pk^{DIAS,U}_{sig}$); The privacy section is the following five-tuple: $Compound^{private}_{U,DIAS}$ = ($pk^{U,DIAS}_{sig}$, $sk^{U,DIAS}_{sig}$, $pk^{DIAS,U}_{sig}$, $sk^{DIAS,U}_{sig}$, $k^{U,DIAS}_{enc}$), which is reserved by the node itself. The following is an example of the private data protection process with a single U and *DIAS*.

B. Store Operation

The core idea of the storage operation is that after encrypting the user's private data, the data sharding by *DIAS* is stored in the *BN* node, and the *SNL* is updated to restore the encrypted private data; *DIAS* also has the same node storage list for sharding, which is stored in the blockchain, and the metadata stored in the list fragment is stored in the *BN* node of the deterministic choice, which is used for the recovery operation of the encrypted private data of the U itself. Only the update value of *SNL* is saved on the blockchain, not the private data itself, which guarantees the security of the data. We use **Protocol 2** Storing Secret Data of User and **Protocol 3** Storing of Metadata to describe these two phases.

TABLE II
RELATED ILLUSTRATION

Element	Illustration	**Algorithm**	Illustration
$pk_{enc}^{A,B}$	The encrypted public key owned by A. After encryption data object faces B.	$\mathrm{E}_F(k,m) \rightarrow c$	Encryption algorithm for fixed algorithm system. Encrypt plaintext data m by using k to get ciphertext c.
$sk_{enc}^{A,B}$	The decryption private key owned by A. The decryption data comes from B.	$\mathrm{D}_F(k \cdot c) \rightarrow m$	The decryption algorithm for fixed algorithm system. Decrypt ciphertext data c with key k, which can obtain plaintext m.
$sk_{sig}^{A,B}$	The signature private key owned by A. The signed data object faces B.	$\mathrm{G}_F(x) \rightarrow y$	Generator for fixed algorithm F. Use x to generate the corresponding data y.
$pk_{sig}^{A,B}$	A's signature public key. The signature data is derived from B.		
$k_{sig}^{A,B}$	Symmetric keys used between A and B that can be used for encryption and decryption.		

Protocol 1 Generate Compound Identity

procedure GenerateCompoundIdentity(*DIAS*, U)
  DIAS, U form a safe channel
  U executes:
  Grandom→r_1, Genc(r_1) → $(pk_{enc}^{U,DIAS}, sk_{enc}^{U,DIAS})$
  Grandom→r_2, Gsig(r_2) → $(pk_{sig}^{U,DIAS}, sk_{sig}^{U,DIAS})$
  Grandom →r_3, Gsym(r_3) → $k_{sym}^{U,DIAS}$
  U shares $pk_{enc}^{U,DIAS}$, $pk_{sig}^{U,DIAS}$ and $k_{enc}^{U,DIAS}$ with *DIAS*
  DIAS executes:
  Grandom→r_4, Genc(r_4) → $(pk_{enc}^{DIAS,U}, sk_{enc}^{DIAS,U})$
  Grandom→r_5, Gsig(r_5) → $(pk_{sig}^{DIAS,U}, sk_{sig}^{DIAS,U})$
  DIAS share $p_{enc}^{kDIAS,U}$ and $k_{sig}^{DIAS,U}$ with U
  U share $pk_{enc}^{U,DIAS}, pk_{enc}^{DIAS,U}, pk_{sig}^{U,DIAS}, pk_{sig}^{DIAS,U}$ and $k_{sym}^{U,DIAS}$ with *DIAS*
  return $pk_{enc}^{U,DIAS}, pk_{enc}^{DIAS,U}, pk_{sig}^{U,DIAS}, pk_{sig}^{DIAS,U}, k_{sym}^{U,DIAS}$
end procedure

Storing Secret Data of User

Protocol 2 Storing Secret Data of User

Procedure StoringSecretData(*DIAS*, U, M, $pk_{enc}^{U,DIAS}$)
  U send the data storage request to *DIAS*
  U and *DIAS* execute Protocol 1 to generate identity and key pairs
  U executes:
  Fix the current secret data M
  Eecc($pk_{enc}^{U,DIAS}$, M) → C, send C to the *DIAS*
  DIAS executes:
  Divide C into the N shares $c_1, ..., c_N$ by using the secret sharing
  Get the available storage nodes information from the bulletin boards
  Select encrypted user data storage nodes $\eta_1^C, ..., \eta_N^C$ randomly from the information from 6)
  Set the restore encrypted user data key as Gsym (H(c_i)) = k_i^C for each shares c_i
  Send k_i^C to the encrypted user data storage node η_i^C
  Each encrypted user data storage node η_i^C executes:
  Esym (k_i^C, c_i) →χ_i , safely stores χ_i and delete k_i^C
  DIAS executes:
  Set the metadata E consisting of $pk_{enc}^{U,DIAS}$ and the pairs (η_1^C, k_1^M), ..., (η_1^C, k_N^M)
  return E
end procedure

Annotation-I:
1) Step 6: N is an arbitrary positive integer, $1 < i \leq N$.
2) Step 10: If the storage node η_i^C already stores data corresponding to the same key k_i^C, η_i^C rejects the query. In this case, return to step 6) .

Storing of Metedata

Protocol 3 Storing of Metadata

Procedure StoringMetadata(U, *DIAS*, E)
  DIAS executes:
  Gpassword →PASSWORD // ID and PASSWORD generate the pair (ID, PASSWORD)
  Get all the SNL from the BN: $L(s_1)$, ..., $L(s_\lambda)$, and select a $L(s_\tau)$ determinately
  Gsym (H(ID, PASSWORD , τ)) → $k_{sym,onetime}^{DIAS,BN}$
  Esym ($k_{sym,onetime}^{DIAS,BN}, E$) → Q
  Divide Q into the ν shares q_1, ..., q_ν by using the secret sharing
  Select the metadata storage nodes ξ_1^E, ..., ξ_ν^E from the selected list $L(s_\tau)$
  Set the metadata storage key as Gsym (H(ID, PASSWORD, τ, n))→ k_n^E for each shares q_n
  Send the metadata storage key k_n^E to the storage node ξ_n^E for each 1 $< n \leq \nu$. Each metadata storage node ξ_n^E executes:
  Esym (k_n^E, q_n) → θ_n, safely stores θ_n and delete k_n^E
  DIAS executes:
  DIAS send ρ→Esym($k_{sym}^{DIAS,U}$, (ID, PASSWORD, τ, ν)) to U
  DIAS use pipeline-based protocol Blockchain storage protocol to publish updated $L(s_\tau)$ to the *Blockchain*
  Send the block height to U
  Delete all the imformation about secret data about U
  return updated *Blockchain*
end procedure

Annotation -II:
1) Step3: λ is the number of all the lists, $1 < \tau \leq \lambda$. s_τ is chosen deterministicly by the time stamp τ as a term, we highly recommend the algorithm as $s_\tau = H(\tau$, PASSWORD) mod $\lambda + 1$.
2) Step4: $k_{sym}^{DIAS,BN}$, onetime is the symmetric key for encryption of the metadata E for one time.
3) Step9: If the metadata storage node ξ_n^E already stores data corresponding to the same key k_n^E, ξ_n^E rejects the query. In this case, return to step 3.

C. Synthetic Decryption Data Operation

The restore operation of the user data operation is in the opposite order of the storage operation, which is divided into two operational phases in a similar manner to the storage operation. First, the corresponding data sharding is found in the *BN* node to restore metadata E. In the latter part, we use the metadata E to find the user private data sharding from the corresponding *BN* node and restore the original private data. We describe these two phases with **Protocol 4** Restoring of metadata and **Protocol 5** Restoring of User Data.

Restoring of metadata:

Protocol 4 Restoring of metadata

Procedure RestoringMetadata(*DIAS*, *Blockchain*, ρ)
U executes:
Send ρ, the block height and restoring of metadata request to *DIAS*
DIAS executes:
Dsym ($k_{sym}^{DIAS,U}$, ρ) $\rightarrow$ (ID, PASSWORD, τ, ν)$'$
Get the storage node list $L(s_\tau)'$ from the *Blockchain* at the block height
Select the metadata storage nodes $\xi_1^{E\prime}, \ldots, \xi_\nu^{E\prime}$ from the list $L(s_\tau)'$
Send the request to $\xi_n^E (1 < n \leq \nu)$ for metadata
Each metadata storage node $\xi_n^{E\prime}$ executes:
Send θ_n' to *DIAS*
DIAS executes:
Compute the metadata restore key $k_n^{E\prime} \rightarrow$ Gsym (H((ID, PASSWORD, τ, n)$'$)) to each metadata storage nodes $\xi_n^{E\prime}$
Dsym ($k_n^{E\prime}$, θ_n') $\rightarrow q_n'$
Reconstruct the metadata Q from each metadata shares from the shares of Step9)
Genc(H(ID, PASSWORD, τ)) $\rightarrow k_{enc}^{DIAS,Blockchain\prime}$
Denc ($k_{enc}^{DIAS,Blockchain\prime}$, Q) $\rightarrow E'$
return E'
end procedure

Annotation -III
Step 7: If the reconstruction of E' fails, it means restoring operation fails.Return back to Step 4)

Restoring of Secret Data of User:

Protocol 5 Restoring of Secret Data of User

Procedure RestoringUserData (*DIAS*, U, E')
DIAS executes:
Parse E' into $(\eta_1^C, k_1^M)'$, ..., $(\eta_1^C, k_N^M)'$
Send the request to each storage nodes $\eta_i^{C\prime}$ $(1 < i \leq N)$ for encrypted user data
Each encrypted user data storage node $\eta_i^{C\prime}$ executes:
Send χ_i' *to DIAS*
DIAS executes:
Dsym ($k_i^{C\prime}$, χ_i') $\rightarrow c_i'$
Reconstruct the ciphertext C' from the shares restored from Step 5)
Send C' to U
U executes:
Decc($p_{enc}^{kU,DIAS}$, C') $\rightarrow M'$
Refix M' to get *User Secret's Data*
return *User Secret's Data*
end procedure

Annotation -IV
1) Step 6 : If the reconstruction of C' fails, it means restoring operation fails.Return back to Step 3)

V. Security Analysis

The components of the comprehensive analysis architecture have three main targets for attackers: (1) Data stored by nodes in *BN*. (2) Authentication information stored by the user. (3) *BN* network decision-making process. We will analyze the security of the three attack targets.

A. Security of Data Stored by BN Nodes

This section discusses security of data stored by *BN* nodes. In the scheme proposed by this paper, user data and metadata are encrypted using a different key, which is stored in the *BN* node by secret sharing, and the storage node is a randomly selected node. Therefore, if an attacker wants to steal user private data or metadata, the following four steps must be fully implemented. Test the target storage node of the storage data sharding; Decrypt the corresponding data sharding from the target node; Reconstructing known ciphertext data; Brute force decrypt ciphertext to get user data or metadata. Based on the four attack steps, we discuss the security of each step.

1) Test the target storage node of the storage data sharding: The *BN* network nodes that store user private data fragments are randomly selected. So, without a metadata storage list $L(s\tau)$, it is difficult for an attacker to retrieve a user's private data storage node. On the other hand, the complexity of the combination of collision attack metadata storage nodes from the storage node list is determined by the user authentication information (ID, PASSWORD, τ). Therefore, if the attacker has no user authentication information, it is difficult to detect the target storage metadata node sharding.

2) Decrypt the corresponding data sharding from the target node: According to the protocol process, if the node wants to recover the corresponding private data sharding from the storage node, the corresponding key must be provided. Because the private data sharding in this paper is the corresponding key is the hash value of the data sharding itself and the hash function has the original root stability, the attacker must provide the data sharding itself to compute the key. The metadata is divided into data slices and hidden in the *BN* network nodes. The attacker must know the location of the decryption key corresponding to the metadata subsection and the location of the node where the metadata is stored to obtain the metadata. The decryption key and location information are determined by the user authentication information (ID, PASSWORD, τ). Therefore, the attacker cannot retrieve the metadata of the metadata without user authentication information, cannot recover the metadata, and cannot restore the private data.

3) Reconstructing known ciphertext data: When an attacker gains partial data sharding, the user has the risk of reconstructing the encrypted message or metadata of the user's private data. In order to reduce the risk of reconstructing ciphertext, we give the following suggestions: it is recommended to use Blakley's threshold secret sharing scheme to carry out data sharding by setting large threshold value to reduce the risk of data reconstruction; And expand the randomness of the private data sharding, the most likely to suppress the possibility of the attacker getting the correct data sharding.

4) Brute force decrypt ciphertext to get user data or metadata: In the scheme designed in this paper, two data shards of Protocol 2 and Protocol 3 are encrypted with different keys. Private and private key pairs of encrypted user private data are owned by U privately; The metadata encryption key consists of a random symmetric key consisting of the *DAIS* one-time password and timestamp. Therefore, the attacker cannot decrypt the ciphertext of the metadata without the user authentication information tuple (ID, PASSWORD, τ). Even if $L(s\tau)$ that corresponds to the metadata is encrypted by brute force, *DIAS* can dynamically update metadata fragmentation schemes to disable old metadata. By regularly updating the metadata storage scheme, it can greatly reduce the risk of

violent cracking of ciphertext.

From what has been discussed above, we know, based on the two sharding processes to protect user private data, user data is strictly to protect the safety of *BN*, it is difficult for the attacker to get the user private data from the *BN*.

B. Security Verification of the Authentication Information Tuple

The attack risk of the authentication information tuple (ID, PASSWORD, τ) is discussed in this section. In this paper, the proposed scheme, τ is used to retrieve the dynamic parameters of SNL list, so the core of data security is to protect user ID and PASSWORD. In view of the security of user ID, this paper proposes using the composite identity algorithm **Protocol 1**. When U interacts with different *DIAS*, it can be used to generate different identities and public and private key pairs. Also, different public keys are used to encrypt the different transfer objects of the authentication information tuple (ID, PASSWORD, τ). In order to solve the problem of weak PASSWORD complexity, this paper puts forward the adoption of τ auxiliary safety information as an independent personal information, increase the complexity of the PASSWORD. If the user dynamically changes the selected storage node, τ will update dynamically. And after updating the corresponding data storage scheme through metadata, all the original metadata cannot be used. Even if the attacker knows the target user ID and PASSWORD, the attacker must detect the correct data storage node before the next timestamp update, and the cost of this attack is huge.

C. Security of BN Network System Decision-Making Process

Based on the multi-party authentication of *BN*, we acquiesce to the existence of illegal nodes exists in *BN* network and illegal nodes participate in the multi-party decision-making process of blockchain. Therefore, each node n in *BN* will depend on their computing resources to determine the weight of voting with a certain probability. This evaluation system provides the possibility of Sybil attack, so we consider the overall security of *BN* network by defining a dynamic trust measurement based on node behavior. We can set the trust of each node to the expected value of its future behavior. It is assumed that each node has a correct evaluation of the binary random variable with probability p for receiving the block, and the mathematical expectation of the correct evaluation block is also p. By estimating this probability by calculating the number of good and bad behaviors of a node, the sigmoid function is used to compress it into a probability $trust_n^{(i)} = \frac{1}{1+e^{-\alpha(\#trust-\#untrust)}}$.

Based on this function, as long as most nodes in *BN* are legitimate security nodes, the evaluation results have high credibility. In order to control the voting weights of non-legal nodes, more weights can be given to secure *DIAS* nodes and legitimate user nodes. By voting weights, increasing the reputation of such nodes and stimulating more *BN* points to execute the correct multi-party decision results to reduce the possibility of Sybil attack.

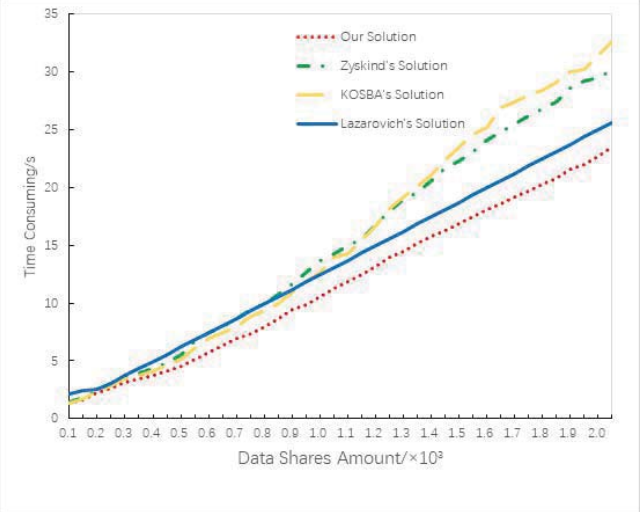

Fig. 3. Data synthesis consumption time comparison

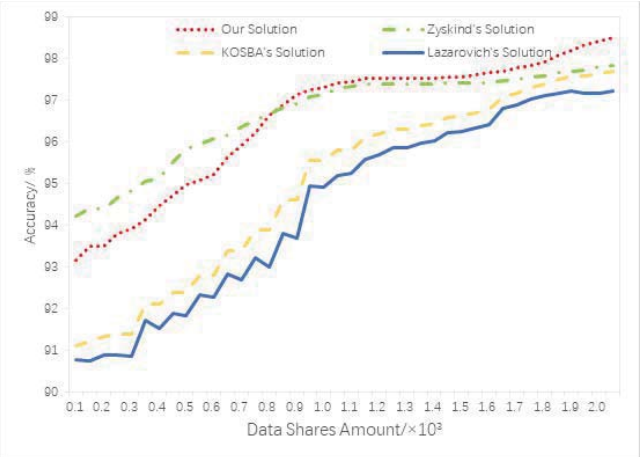

Fig. 4. Data synthesis accurary comparison

VI. Experiment

The experiment of this section mainly shows the performance of the core components of the program, mainly including the following two aspects. 1) The data synthesis efficiency test of the scheme designed in this paper. 2) The accuracy test of synthetic data designed in this paper. The experiment was deployed on 10 physical servers and on a desktop computer. Each server was equipped with two separate network CARDS, each with 250 virtual IP addresses. Set the virtual IP address to the ID of the corresponding node (a total of 5000 nodes), and initialize the operation, and the IP address as the index file store name. The server is configured as follows: the processor was Intel® Xeon® Processor E5-2670 (20M Cache, 2.60GHz, 8.00 GT/s Intel® QPI) Double CPU; 8 GB of internal memory; The operating system was Ubuntu 16.04.4 LTS server, and the kernel version number was 4.4.0-31-generic. 10 servers were named node1, node2, ..., node10. The private data provider is the node in node10.

All nodes of the network are set to *BN* nodes. The *BN* protocol layer selects OpenZeppelin open source architecture as the underlying intelligent contract framework. Private data tested was the medical data in the literature [7] as the case.

A. Program Data Synthesis Efficiency Test

The program data synthesis efficiency test mainly detects the relationships among synthetic performance, test run time and the size of the shardings of the scheme splitting the private data under the same privacy protection environment. For this purpose, the test data set was divided into different sized data shards for gradient testing. the test was divided into 200 shards, and the test private data was fixed at 1MB. And they were compared with the private data protection scheme of literature [6]. As can be seen from Figure 3, the method proposed in this paper significantly reduces the operation time of the original private data.

B. Program Data Synthesis Accuracy Test

The test was mainly about the relationships among the data of the private data and the original data and the accuracy and the size of the shards under the same privacy protection environment. The private data of this test was fixed to 1MB, and the test interval was 200 pieces, using gradient test and comparing the results with the private data protection scheme of literature [5] [6] [7]. It can be seen from Figure 4 that the accuracy of the scheme is slightly lower than the literature [6]. Compared with the literature [6], the number of shards in the paper has been improved compared with that of literature [6], which proves that the scheme is more suitable for the implementation of large-scale network environment. We believe that the reason for the lower accuracy of the proposed scheme designed in this article compared with that recorded in paper [6] under the premise of less than 800 of slices is that, in the case of a small number of slices, there is repetition and redundancy in individual slices, resulting in the relatively low efficiency of the design in this paper.

VII. Conclusion

In this paper, we propose a new scheme for online storage private data without TTP, propose a data protection algorithm in the process of secondary sharding, and implement the blockchain based on pipeline system in this paper. The proposed scheme adopts the idea of composite identity encryption and multi-party data sharing, which divides the storage list of users' privacies into protection, realizes the possibility of the enemy's minimum attack, and can dynamically update the storage scheme with high anti-aggressiveness. It is the next step for us to evaluate and formalize the security quantification based on the proposed scheme.

References

[1] James Ball. Nsa's prism surveillance program: how it works and what it can do. The Guardian, 2013.

[2] Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash Symtem[OL]. http://bitcoin.org/bitcoin.pdf, 2008

[3] Vindu Goel. Facebook tinkers with users' emotions in news feed experiment, stirring outcry. The New York Times, 2014.

[4] Swan M. Blockchain Thinking : The Brain as a Decentralized Autonomous Corporation [Commentary][J]. IEEE Technology & Society Magazine, 2015, 34(4):41-52

[5] Zyskind G, Nathan O, Pentland A'. Decentralizing Privacy: Using Blockchain to Protect Personal Data[C]// IEEE Security and Privacy Workshops. IEEE, 2015:180-184.

[6] Kosba A E, Miller A J, Shi E, et al. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts[C]. ieee symposium on security and privacy, 2016: 839-858.

[7] Lazarovich A. Invisible Ink: blockchain for data privacy[D]. Massachusetts: Massachusetts Institute of Technology, 2015:36-40.

[8] A. Shamir, "How to Share a Secret, " Commun. ACM vol. 22, pp. 612–613, 1979.

[9] Fukumitsu M, Hasegawa S, Iwazaki J, et al. A Proposal of a Secure P2P-Type Storage Scheme by Using the Secret Sharing and the Blockchain[C]// IEEE, International Conference on Advanced Information NETWORKING and Applications. IEEE, 2017.

[10] L. Yu, et al, Smart-Contract Execution with Concurrent Block Building, 2017 IEEE Symposium on Service-Oriented System Engineering (SOSE).

[11] M. Laskowski, et al, Rapid Prototyping of a Text Mining Application for Cryptocurrency Market Intelligence, 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI).