Zheng BK, Zhu LH, Shen M *et al.* Scalable and privacy-preserving data sharing based on blockchain. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 33(3): 557–567 May 2018. DOI 10.1007/s11390-018-1840-5

Scalable and Privacy-Preserving Data Sharing Based on Blockchain

Bao-Kun Zheng[1,2], Lie-Huang Zhu[1], *Member*, *CCF*, *IEEE*, Meng Shen[1,*], *Member*, *CCF*, *IEEE*, Feng Gao[1] Chuan Zhang[1], Yan-Dong Li[1], and Jing Yang[1]

[1] *School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China*

[2] *School of Information Management for Law, China University of Political Science and Law, Beijing 102249, China*

E-mail: zhengbk168@163.com; {liehuangz, shenmeng}@bit.edu.cn; gaofengbit@foxmail.com
{chuanz, leeyandong, jingy}@bit.edu.cn

Received November 20, 2017; revised March 28, 2018.

Abstract With the development of network technology and cloud computing, data sharing is becoming increasingly popular, and many scholars have conducted in-depth research to promote its flourish. As the scale of data sharing expands, its privacy protection has become a hot issue in research. Moreover, in data sharing, the data is usually maintained in multiple parties, which brings new challenges to protect the privacy of these multi-party data. In this paper, we propose a trusted data sharing scheme using blockchain. We use blockchain to prevent the shared data from being tampered, and use the Paillier cryptosystem to realize the confidentiality of the shared data. In the proposed scheme, the shared data can be traded, and the transaction information is protected by using the (p, t)-threshold Paillier cryptosystem. We conduct experiments in cloud storage scenarios and the experimental results demonstrate the efficiency and effectiveness of the proposed scheme.

Keywords data sharing, privacy-preserving, cloud computing, blockchain, Paillier cryptosystem

1 Introduction

With the development of network technology, the amount of personal data grows rapidly[1]. To store and share the data, it is attractive to shift the data storage and sharing applications into the cloud for resources and economic savings[2], which however brings challenges to information security, such as data loss and privacy leakage[3]. Moreover, by storing the data in the cloud, people loss full control to their personal data, which makes it essential to ensure the confidentiality, integrity and privacy of the data. Besides the above challenges, in some data sharing applications, the data is usually required to be maintained by multiple parties, such as in certification authorities to protect the private root certificate keys[4-6]. How to solve the information security problem under this new situation is still an urgency and one of the most concerned issues in the field of data sharing[7].

Inspired by the idea of shared economy[8], we can imagine such a scenario that users with similar data can conduct data transactions according to their needs or get meaningful results through computation. They can make an alliance without a trusted center authority (CA), which is supported by a novel technology, called blockchain①. Blockchain inspires us that data sharing can maintain a tamper-proof ledger shared by the participating users without the need of a trusted third party. In this way, data sharing can be effectively integrated and utilized. In our scenario, users can use blockchain to prevent the shared data from being tampered, and can use blockchain to carry out transactions that can be tracked.

Using cryptographic techniques is powerful to preserve the privacy of the sensitive data. In 1979, Shamir[9] and Blakley[10] proposed a (p, t)-threshold

Regular Paper
Special Section on Blockchain and Cryptocurrency Systems
*Corresponding Author
①Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf, Mar. 2018.

secret sharing scheme designed to protect the secret by distributing a secret among a group of p participants. The ciphertexts can be decrypted accurately only if more than t participants cooperate with one another. The (p, t)-threshold secret sharing scheme has been used in various applications, such as certification authorities to protect the private root certificate keys[4-6].

In this context, it is very meaningful to study the privacy security research of data sharing based on the combination of the blockchain and (p, t)-threshold encryption technology. To solve the above-mentioned challenges, we propose a scheme to implement scalable and privacy-preserving data sharing based on blockchain. The main purpose is to realize the privacy protection of data shared by multiple parties. To summarize, the contributions of our work are as follows.

1) We propose a new scalable and privacy-preserving data sharing scheme based on blockchain. In the proposed scheme, blockchain is used to prevent the shared data from being tampered, and carry out transactions which can be tracked. The data is encrypted and stored in the cloud, and the key is allocated by CA. Moreover, in the scheme, the shared data can be traded, and the transaction information is encrypted among multiple users to ensure its security and reliability.

2) We propose a new method to protect multi-party data privacy in blockchain. In this new method, the (p, t)-threshold Paillier cryptosystem is applied to the blockchain. The private key sk is separated (denoted as $sk_1, sk_2, ..., sk_p$) and distributed to p participants. If one party wants to decrypt the ciphertext c, at least $(t-1)$ private keys of other parties are needed to be aggregated.

3) We analyze the security of the proposed scheme, and the analysis proves that our scheme can capture the privacy preservation. Moreover, performance evaluation via extensive experiments demonstrates the efficiency and effectiveness of the proposed scheme.

The remainder of this paper is organized as follows. Section 2 introduces preliminaries of our work. Section 3 gives a concrete description of our method model. The detailed design of privacy-preserving protocol is presented in Section 4. Section 5 and Section 6 carry out privacy and performance analysis respectively. Section 7 summarizes related work. Section 8 concludes this paper.

2 Preliminary

In this section, we introduce blockchain, homomorphic encryption, and the Paillier cryptosystem because they are important components of our proposed scheme.

2.1 Blockchain

Blockchain is essentially a peer-to-peer (P2P) network distributed ledge database, which is a series of data blocks generated by cryptography. Each transaction is approved by the majority of the participants in the system[6]. A complete blockchain system contains a lot of technologies, such as a P2P network, distributed ledger, asymmetric encryption, consensus mechanism, and smart contract[11]. It is these technologies that make the blockchain a continuous engine on distributed networks, providing a steady flow of power to the transaction validation links on the blockchain.

Blockchain has the characteristics of decentration, timing data, collective maintenance, programmability, security, trust, etc.[11]

1) *Decentralization.* The validation, bookkeeping, storage, maintenance, and transmission of blockchain data are based on a distributed system structure. The blockchain uses pure mathematical methods instead of central institutions to establish trust relations among distributed nodes, thus forming a decentralized and trustworthy distributed system.

2) *Timing Data.* Blockchain stores data with a time-stamped block structure. Thus, it adds time dimension to data, and has extremely strong verifiability and traceability.

3) *Collective Maintenance.* The blockchain system adopts specific economic incentive mechanism to ensure that all nodes in the distributed system can participate in the verification process of data blocks (such as bitcoin mining process). The new block is added to blockchain through the consensus algorithm.

4) *Programmability.* Blockchain can provide flexible script code system, and support users to create advanced smart contracts, currencies or other decentralized applications. For example, Ethereum② that provides Turing-complete script language for the user to build any smart contract or transaction type can be precisely defined.

5) *Security and Trust.* Blockchain adopts asymmetric cryptography principle to encrypt data. The consen-

②https://github.com/ethereum/wiki/wiki/White-Paper, Mar. 2018.

sus algorithm forms a computing power to resist external attack, and prevent the blockchain data from being tampered. Thus, blockchain has higher security.

2.2 Homomorphic Encryption and Paillier Cryptosystem

Homomorphic encryption is first proposed by Rivest *et al.*[12] in 1978. Homomorphic encryption allows users to directly perform algebraic operations specific to ciphertext, get results, and perform the same operation on the same plaintext encrypted result. The public key pk and the private key sk are generated by the security parameter λ. pk is used to encrypt plaintext and sk is used to decrypt the ciphertext. Supposing a plaintext $m \in Z_n$, where n is a large positive integer and Z_n is the set of integers modulo n, we denote the encryption of m as $E_{pk}(m)$. Homomorphic encryption has the properties

$$E_{pk}(m_1 + m_2) = E_{pk}(m_1) \bigoplus E_{pk}(m_2),$$
$$E_{pk}(a \times m_1) = a \bigotimes E_{pk}(m_1),$$

where m_1 and m_2 are the plaintexts that need to be encrypted and a is a constant.

The Paillier cryptosystem is a kind of encryption systems based on the high order residue class problem proposed by Pailler in 1999[13]. The homomorphism properties[14] of the system can be used to construct many practical and efficient cryptographic algorithms. In this paper, we mainly study to protect multi-party data privacy, thereby the threshold Paillier cryptosystem is used in our scheme, as it not only has additive homomorphic properties but also satisfies the design of a threshold cryptosystem.

In this paper, we adopt the (p, t)-threshold Paillier cryptosystem, in which the private key sk is divided (denoted as $sk_1, sk_2, ..., sk_p$) and distributed to p parties. Each party has an incomplete private key. If a party wants to decrypt the ciphertext, at least $(t-1)$ parties are needed.

Specifically, in the step of decryption, each party i $(1 < i < p)$ is required to compute its partial decryption c_i by using its private key sk_i, as

$$c_i = c^{2\Delta sk_i}, \tag{1}$$

where $\Delta = p!$. Then based on the combining algorithm in [12], at least t partial decryptions can be combined together to get the plaintext m.

3 Scalable Storage and Privacy-Preserving Model

In this section, we first formalize the entities and operations involved in our scalable storage and privacy-preserving model in blockchain. Second, we give the privacy assumption, and then show the design objectives.

3.1 System Model

Fig.1 shows our system model. Entities in our model are as follows.

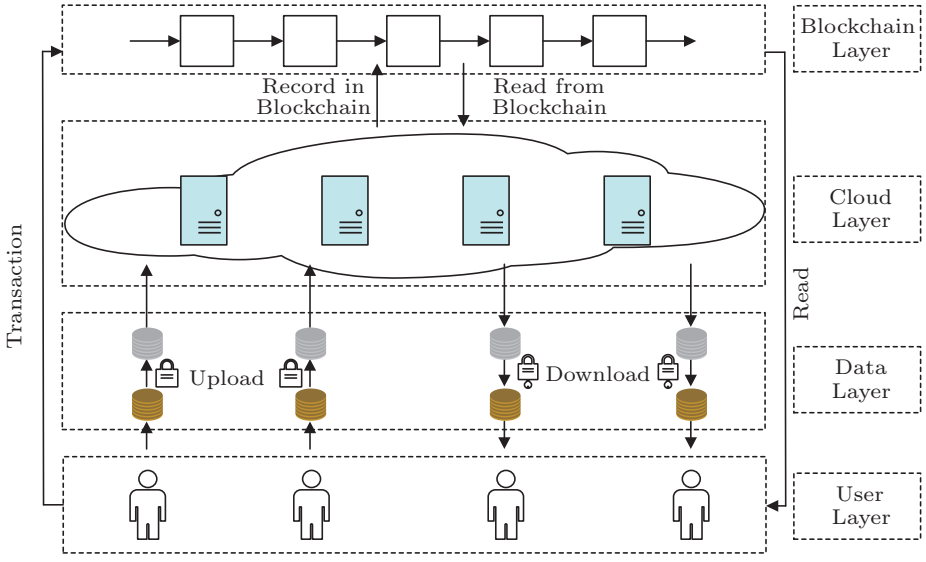


Fig.1. Scalable storage and privacy-preserving model.

User Layer. Users in the user set have the same kind of data, and they trade data through data sharing, or get meaningful results by computing. Users can use blockchain to prevent the shared data from being tampered, and can further use blockchain to carry out transactions that can be tracked.

Data Layer. Data represents information that users want to protect and share. Users in the user set can collectively maintain the data. To protect data privacy, the data is encrypted using the Paillier cryptosystem. After that, the encrypted data is uploaded to the cloud for sharing. Because the threshold homomorphic cryptosystem is used, users in the user set can perform calculations on the data to decrypt the ciphertexts if needed.

Cloud Layer. The cloud is used to store encrypted data from users. The cloud provides encrypted data for uploading and downloading services, as well as reading and writing interactions with the block chain.

Blockchain Layer. Blockchain provides a powerful abstraction for the design of distributed protocols. The cloud can write contents into the blockchain and read the contents from it. Once a block is collectively accepted, it is practically impossible to change it or remove it, which is guaranteed by the nature of the blockchain[11]. User data can also be traded through the blockchain. In our scheme, the transaction information is encrypted using the Paillier cryptosystem, and it can be decrypted with the consent of the user.

3.2 Privacy Assumption

In this paper, we adopt two assumptions: there are at most $(t-1)$ parties colluding with one another and all the parties are semi-honest[14]. The collusion model assumes a user cannot collude with other more than $(t-1)$ users, which means a user cannot decrypt the ciphertexts by colluding with other parties. The semi-honest model assumes that all parties are honest but curious, which means they strictly abide by our protocol design, but each party will try to deduce the private information of other parties during the execution of the protocol. These assumptions are reasonable in most blockchain transaction scenarios, since 1) all parties wish to get the right results and abide by the common benefits of the protocol, and 2) users usually do not know one another, and even if they know one another, they may not want to divulge their private information to others.

Before entering the details of our privacy requirements, we observe that the main purpose of blockchain is to protect privacy. More importantly, we aim to protect data privacy and transaction information privacy at the same time. The definition of data privacy and transaction information privacy is given below.

Definition 1 (Threshold). *Suppose there are p users, denoted as $p = \{1, 2, ..., P\}$, and t users, represented as $t = \{1, 2, ..., T\}(t \leqslant p)$. At least, t users can decrypt the encrypted data. In this procedure, each value should be confidential to any party except the user who provides this value.*

Definition 2 (Data Privacy). *Some user data which occupies a large amount of storage space will be stored in the cloud, but the hash value of the data is stored on the blockchain. The data stored in the cloud is encrypted using the Paillier cryptosystem. The key is distributed to p users, and each user has only one part of the key, and a threshold user key combination can be used to decrypt the data, so as to ensure the privacy of user data.*

Definition 3 (Transaction Information Privacy). *When user A wants to send some coins to user B, user A does not want anyone to know with certainty that the coins go to user B. The transaction information is encrypted using the Paillier cryptosystem, so that other users cannot know the information. Thus, the privacy of the transaction information is protected.*

3.3 Design Goals

In this paper, we mainly aim to protect data privacy and transaction information privacy as defined in Subsection 3.2. Meanwhile, it is necessary that the servers can still provide high-quality service when operating on data storage.

Besides the objective on privacy preservation, the system should have the following design goals.

High Accuracy Service. CA completes key distribution and verification with high accuracy.

Low Response Time Service. The cloud completes the data storage, key distribution and verification process in an acceptable period of time.

4 Implementation

In this section, we discuss the details of our novel privacy-preserving scheme.

4.1 Privacy-Preserving Scheme Overview

Fig.1 shows our scalable storage and privacy-preserving scheme in blockchain. We assume that a

semantically secure (p,t)-threshold Paillier cryptosystem has been established by the third party. Here p refers to the number of parties including CA and users, and t is the minimum number of parties required to complete the decryption. Thus, the public encryption key $pk = (g, n)$ is public to each party in this scheme; however, the matching private decryption key sk is divided and distributed to all parties (i.e., party i can get his private key sk_i). In Fig.1, the scheme can be divided into four phases.

Phase 1: *Encrypt and Decrypt User Data.* The user's privacy data is encrypted by the Paillier cryptosystem, CA is responsible for the distribution of privacy keys, and the encrypted data can be decrypted with at least t users.

Phase 2: *Upload and Download Files.* The cloud supports the following operations: upload and download. In addition, the cloud generates data hash values, which are used to generate a proof of data.

Phase 3: *Record in Blockchain and Read from Blockchain.* The cloud can write contents into the blockchain and read the contents from it.

Phase 4: *Transaction.* Users in the user set can trade their shared data through the blockchain. The transaction information is encrypted with the Paillier cryptosystem and it can be decrypted with at least t users.

4.2 Data Privacy-Preserving Protocol

In order to protect the privacy and tamper resistance of user data, it is necessary to encrypt the data. In this paper, since the data is maintained by multiple parties, the encrypted data requires a certain number of users to consent to decrypt it. We adopt the (p,t)-threshold Paillier cryptosystem. The private key sk is separated (denoted as $sk_1, sk_2, ..., sk_p$) and distributed to p users. If one party wants to decrypt the ciphertext c, at least $(t-1)$ private keys of other parities need to be aggregated.

The specific process is as follows.

• *Step* 1. According to

$$c = E_{pk}(m) = g^m r^n \bmod n^2,$$

where the plaintext $m \in Z_n$ with the public key $pk = (g, n)$ and $r \in Z_n^*$ (Z_n^* denotes the multiplicative group of invertible elements of Z_n) are selected randomly and privately by this user. Each user $k \in K$ encrypts value v_k and sends the ciphertext $E_{pk}(v_k)$ to the data center.

• *Step* 2. According to the homomorphic property

$$\begin{aligned} E_{pk}(m_1 + m_2) &= E_{pk}(m_1) + E_{pk}(m_2) \\ &= g^{m_1+m_2}(r_1 r_2)^n \bmod n^2, \end{aligned}$$

where m_1, m_2 are the plaintexts that need to be encrypted, and $r_1, r_2 \in Z_n^*$ are the private random.

Cloud calculates

$$C = E_{pk}(\sum_{k=1}^{K} v_k) = \prod_{k=1}^{K} E_{pk}(v_k).$$

CA selects t users as needed and sends C to them.

• *Step* 3. Each selected user k' calculates the partial decryption $C_{k'}$ of C based on (1) and sends $C_{k'}$ to cloud.

• *Step* 4. Cloud combines it with $(t-1)$ partial decryptions received from users to get the result.

4.3 Data Uploading and Downloading Protocol

The operation between the user and the cloud is to upload and download data.

1) *Uploading Protocol.* The uploading protocol is executed between the cloud and users who aim to upload a file f. Let $H : \{0,1\}^* \to \{0,1\}^l$ be a cryptographic hash function, where l represents the token size. The specification of the uploading procedure is shown as follows.

• *Step* 1. For a file f to be uploaded, $user_1$ first generates a hash key $K_f = H(f)$, where H is a hash function. K_f can identify the local location of the file f.

• *Step* 2. To achieve confidentiality, $user_1$ encrypts the data file f as $f^* = E_{pk}(f)$. $user_1$ then saves K_f in the local place, computes a digital fingerprint $K_f^* = H(f^*)$ of f, and saves it in the local place.

• *Step* 3. $User_1$ sends f^* and K_f^* to the cloud.

• *Step* 4. Upon receiving K_f^*, the cloud will compute $K_f^{**} = H(K_f^*)$.

Each user can upload their own data in this way.

2) *Downloading Protocol.* The specification of the downloading procedure is shown as follows.

• *Step* 1. A client $user_2$ issues a request to cloud to download a file f attached to K_f^{**}.

• *Step* 2. Upon receiving the package (f^*, K_f^{**}), $user_2$ fist computes if $H(H(f^*)) = K_f^{**}$. If not, the user requests. If the equation is established, then $user_2$ can decrypt it to f with sk.

4.4 Record in Blockchain Protocol

1) *Data Block.* To ensure that the record content is trusted and untampered, the data hash value is stored in the item structure we built. The hash value of each item is put into the item block structure, which can effectively reduce the search space and speed up the users' checking speed of the record. The data block is made up of multiple item blocks, and the hash value is calculated to get the Merkle root of the data block. Merkle root is submitted to the blockchain so that the data cannot be tampered. Each item block stores only the hash value of item and a header information, which not only facilitates the propagation of each block in the P2P network, but also reduces the cost of data validation. Each item can store 10 pieces of data, each containing three kinds of information: the data owner public key, the metadata, and the hash value. The storage structure and the item data structure of Merkle tree are shown in Fig.2 and Fig.3 respectively.

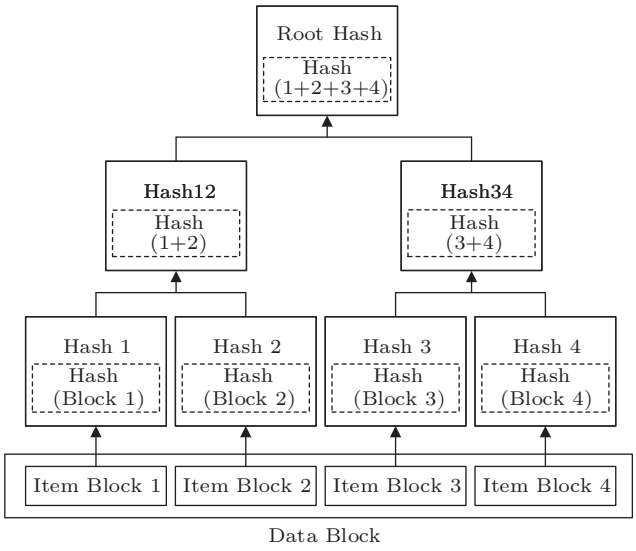


Fig.2. Merkle tree of storage structure.

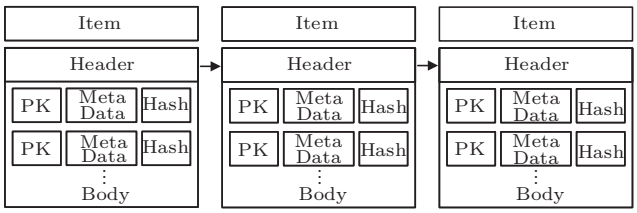


Fig.3. Item data structure.

2) *Smart Contract.* The smart contract is used to write data information to the blockchain. The specification of the smart contract procedure is shown as follows.

• *Step* 1. The user records the request and presents the public key as the identity.

• *Step* 2. A representative node accepts the request.

• *Step* 3. A representative node broadcasts the request that has accepted.

• *Step* 4. The user submits the record.

• *Step* 5. The representative node adds the record to the item according to the users' public key.

• *Step* 6. The representative node broadcasts item validation information.

• *Step* 7. The representative node verifies the record, and other nodes update the data.

• *Step* 8. Check the number of item blocks every 1 minute, up to 10 to make up a data block, and calculate the Merkle root of the block.

• *Step* 9. Anchor the Merkle root of all newly generated data blocks to the blockchain.

• *Step* 10. Return to step 1.

4.5 Transaction Information Privacy-Preserving Protocol

Our protocol also supports the partial anonymity of the transaction by using the blind property of the Paillier cryptosystem. Image that Alice wants to send some coins to Bob, but she does not want anyone else to know that the coins are for Bob. To do this, she chooses a certain number of accounts to form an anonymous set, and supposes the number is n. Then we pick $(n-1)$ random addresses from the anonymity set. Finally, she executes a transfer, where she sends the certain number of coins to Bob's address and zero coins to all the other $(n-1)$ random addresses. Now, the ciphertexts of the balances of the $(n-1)$ random accounts that Alice chooses will change but the actual balances will not because the Paillier cryptosystem has the blinding property.

1) To create a key pair, we first choose two primes, g and n, with an equal length. The encryption key pk will be $N = g \times n$ and the decryption key sk will be $\lambda = (g-1) \times (n-1)$.

2) To encrypt a message $m \in Z_n$ with the public key $pk = (g, n)$, we pick a random integer $r \in Z_n^*$ and compute the ciphertext as

$$E_{pk}(m, r) = (N+1)^m \times r^N \bmod N^2.$$

3) To decrypt, we calculate the original message as

$$m = \frac{(E_{pk}(m, r)^\lambda \bmod N^2) - 1}{N} \times \lambda^{-1} \bmod N.$$

We can also recover the random integer r used in a given ciphertext by the formulas

$$r = c^{N^{-1} \bmod \lambda} \bmod N,$$
$$c = E_{pk}(m, r) \times (N+1)^{-m} \bmod N,$$

according to the homomorphic properties.

$$E_{pk}(m_1, r_1) \times E_{pk}(m_2, r_2) = E_{pk}(m_1 + m_2, r_1 \times r_2),$$
$$E_{pk}(m, r)^k = E_{pk}(k \times m, r^k).$$

Since the Pailier cryptosystem has the blinding property

$$E_{pk}(m, r_1 \times r_2) = E_{pk}(m, r_1) \times E_{pk}(0, r_2),$$

it can change a ciphertext without changing the corresponding plaintext.

5 Privacy Analysis

As discussed before, the privacy threats mainly come from the parties themselves in practical systems. Thus, the goal of our scheme is to protect the data of each user from being disclosed to other parties. Since our scheme is built upon the proposed data privacy-preserving protocol, we analyze the security of the proposed protocol in this section.

In the data privacy-preserving protocol, the data is exchanged only between the data center and users, and all the exchanged data is ciphertexts. Although some users get the ciphertext of summation

$$E_{pk}(\sum_{k=1}^{K} v_k),$$

they cannot decrypt it because we use the (p, t)-threshold Paillier cryptosystem and assume that a user cannot collude with more than $(t-1)$ other users. Therefore, after the protocol is executed, the user will learn nothing. Moreover, the ciphertext $E_{pk}(v_k)$ cannot be decrypted by the data center, and what the data center can know at last is just the summation $\sum_{k=1}^{K} v_k$, based on which it cannot deduce the private key v_k of each user. Therefore, the privacy of each user's private key is guaranteed by this protocol.

Then we can summarize the data privacy-preserving goal of our scheme as follows.

Assume $K \leqslant 3$ and for each part $m \in M$, there are at least two users $k_1, k_2 \in K$ giving different values (i.e., $x_m^{k1} \neq x_m^{k2}$). Also suppose the parties are semi-honest and there is no collusion among them. Then after the execution of the data privacy-preserving protocol, the values of each user will not be disclosed to others.

Next, in the transaction information privacy-preserving protocol, an attacker who knows a transfer will not be able to know whether Alice really transfers coins to a given address. In fact, only Alice knows to which addresses the coin was transferred. Even Bob only knows that he received some coins, and knows nothing about the amount of coins transferred to another address. Therefore, if an attacker knows that Alice transferred coins to the n accounts in the transfer, only the probability of $\frac{1}{n}$ is chosen for the Bob account.

However, if Alice intends to send some transfers to Bob, she needs to always use her anonymous account set used in the first transaction. Otherwise, the attacker will see multiple transactions where Bob's address appears together with other addresses that never appear again (or not often) and will assume that Alice made those transfers to Bob's address while trying to hide it by using different anonymity sets for each transaction.

6 Performance Evaluation

In our implementation, we set the security parameter $n = 256$. All the users are implemented in Java and the experiment is conducted on some desktop computers which are running Windows 10 and equipped with Intel® Core™ I7 processor with 2.40 GHz and 4 GB RAM. As for blockchain, we select the Fabric[③] as the underlying technology of the blockchain-based settlement system, which is an open source permissioned blockchain technique hosted by the Linux Foundation. We use zero-knowledge proof to identify encrypted transaction information in fabric. Our implementation of cloud interfaces with Aliyun servers, which are equipped with Intel® Xeon® processor with 2.60 GHz and 8 GB RAM.

6.1 Evaluation of Key Generation Efficiency

In the Paillier cryptosystem, the public key is (g, n), where $n = p \times q$, and the private key is

$$\lambda = \mathrm{lcm}((p-1), (q-1)).$$

In our experiments, we generate 64, 128, 256, 512 and 1 024 bits keys and corresponding g and λ respectively to calculate the public key and the private key. As

[③] Hyperledger Fabric. https://arxiv.org/pdf/1801.10228v1.pdf, Mar. 2018.

the length of the generated key grows gradually, the amount of computation required in the generation process increases gradually, and the increase rate increases gradually as shown in Fig.4.

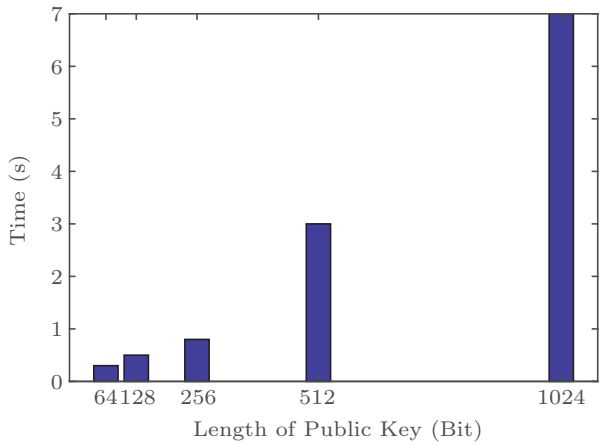


Fig.4. Paillier key generation time.

6.2 Evaluation of Encryption and Decryption Efficiency

Fig.5 shows the time spent encrypting 32 bits data in the case that the public key n is 64, 128, 256, 512 and 1024 bits respectively. As shown in Fig.5, with the increase of the length of the key n, the time consumed in the process of encryption and decryption is gradually increasing, and the amount of computation in the process of encryption and decryption is increasing gradually too.

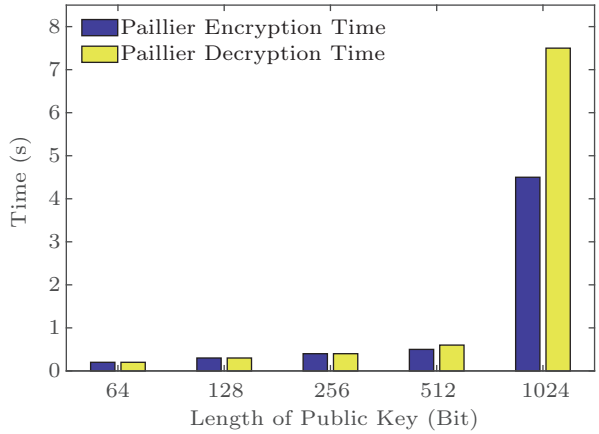


Fig.5. Paillier encryption and decryption time.

6.3 Efficiency Analysis of Homomorphic Properties of the Paillier Cryptosystem

The public key n is 64, 128, 256, 512 and 1 024 bits respectively, and the size of data m is 32 bits. Under the same conditions, we evaluate the time consumption of homomorphic properties of the Paillier public key cryptosystem, which are listed as follows:

1) $E_{pk}(m, r)^k$,

2) $E_{pk}(m_1, r_1) \times E_{pk}(m_2, r_2)$,

3) $E_{pk}(m, r_1 \times r_2)$,

4) $E_{pk}(a \times m_1)$, and

5) $E_{pk}(m_1 + m_2)$.

The evaluation results are exhibited in Fig.6. The results show that additive homomorphism is more time-consuming than multiplicative homomorphism.

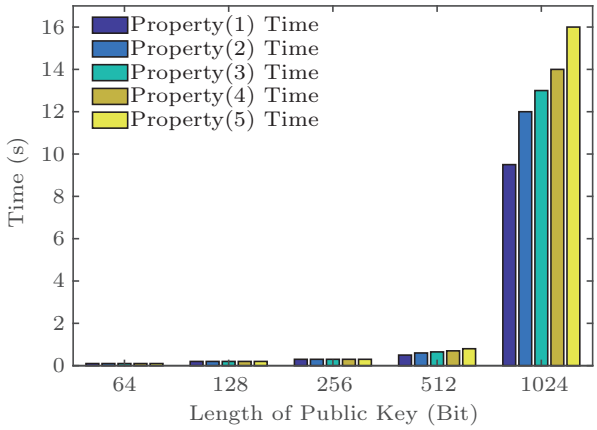


Fig.6. Paillier homomorphic properties time.

6.4 Relation Between Confirmation Time and Number of Concurrent Transactions

Fig.7 depicts the relation between confirmation time and the number of concurrent transactions per second. We run the test 30 times and record the time-consuming average as the result. Note that when the volume of concurrent transactions is relatively small, the system has to wait for the predefined batch time to pack a block. When the number of concurrent transactions per second ranges from 100 to 300, the transaction confirmation time decreases with the increase of concurrent transactions. This is because the number of transactions is reaching the threshold of quantity to pack a block in Fabric. When the transaction volume exceeds the processing capability, some transactions cannot be confirmed in a timely manner which causes that the transaction confirmation time begins to grow.

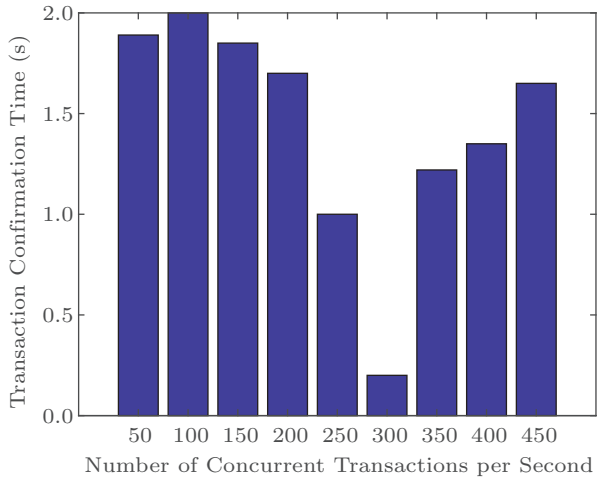


Fig.7. Relation between confirmation time and the number of concurrent transactions.

7 Related Work

The related work can be classified into two categories: data storage via blockchain and the Paillier cryptosystem.

7.1 Data Storage via Blockchain

In order to solve the problem of blockchain storage capacity, some researchers have put forward some solutions. Gaetani *et al.*[15] designed and implemented the cloud computing environment database system based on blockchain, using blockchain to ensure data integrity, improve the performance, and enhance the stability. Ali *et al.*[16] proposed the Blockstake naming storage system, and the system where the four-tier architecture is designed, which fully utilized the decentralization characteristics of the blockchain to ensure the high security of the data. Liang *et al.*[17] proposed a distributed and trusted cloud data traceability using blockchain. Storage-oriented blockchains like Filecoin[18], Storj[18], and Permacoin[19] store distributed files into blocks as a series of transactions.

7.2 Paillier Cryptosystem

A few studies focus on protecting the blockchain privacy problem by using the Paillier cryptosystem. The Paillier cryptosystem is a way to divide a secret information into serval parts that can be given to different parties, with two properties.

1) Any part of information can reconstruct the secret, as long as the size of the part equals or exceeds a specified threshold.

2) Any part of the information whose size is less than this threshold will not get information about this secret.

Frikken[20] proposed a scheme of threshold signature. A key characteristic of threshold signature is that the private key does not need to be rebuilt. Even after multiple signatures, no one knows any information about the private key, which allows users to generate signatures without threshold size groups. Threshold cryptography is a special case, which can make secure multi-party computation get more extensive development. Shamir[9] showed how to divide data D into n pieces in such a way that D is easily reconstructable from any k pieces, but even complete knowledge of $(k-1)$ pieces reveals absolutely no information about D. In most schemes, the secret information can be encoded as a $(t-1)$-order polynomial, a random point on the polynomial is given to each of n parties, and any t of the parties can be used to accurately rebuild the polynomial using Lagrange interpolation. Fouque *et al.*[21] and Baudron *et al.*[22] proposed a multi-party threshold Paillier cryptosystem scheme. The scheme solves the problem of malicious attack, and is proved to be complex and elusive.

8 Conclusions

In this paper, we proposed a scalable and privacy-preserving data sharing scheme based on blockchain. The Paillier cryptosystem was applied to blockchain, which could effectively protect sensitive information and solve the privacy protection problem of blockchain. The security analysis proved that the proposed scheme holds high security. We conducted experiments in cloud storage scenarios. The experimental results demonstrated that the Paillier cryptosystem homomorphism has high efficiency in ciphertext operations.

In the future, we will focus on improving system efficiency and exploring the application of this method in different fields.

References

[1] Sharma S. Expanded cloud plumes hiding big data ecosystem. *Future Generation Computer Systems*, 2016, 59: 63-92.

[2] Yu Y, Ni J B, Man H A, Mu Y, Wang B Y, Li H. Comments on a public auditing mechanism for shared cloud data service. *IEEE Trans. Services Computing*, 2015, 8(6): 998-999.

[3] Kandukuri B R, Ramakrishna P V, Rakshit A. Cloud security issues. In *Proc. IEEE International Conference on Services Computing* (*SCC*), September 2009, pp.517-520.

[4] Stinson D R. An explication of secret sharing schemes. *Designs, Codes and Cryptography*, 1992, 2(4): 357-390.

[5] Beimel A. Secret-sharing schemes: A survey. In *Coding and Cryptology*, Chee Y M, Guo Z B, Ling S, Shao F J, Tang Y S, Wang H X, Xing C P (eds.), Springer, 2011, pp.11-46.

[6] Peng K. Critical survey of existing publicly verifiable secret sharing schemes. *IET Information Security*, 2012, 6(4): 249-257.

[7] Maheshwari N, Kiyawat K. Structural framing of protocol for secure multiparty cloud computation. In *Proc. the 5th Asia Modelling Symp.* (*AMS*), July 2011, pp.187-192.

[8] Hamari J, Sjöklint M, Ukkonen A. The sharing economy: Why people participate in collaborative consumption. *Journal of the Association for Information Science and Technology*, 2016, 67(9): 2047-2059.

[9] Shamir A. How to share a secret. *Communications of the ACM*, 1979, 22(11): 612-613.

[10] Blakley G R. Safeguarding cryptographic keys. In *Proc. AFIPS 1979 National Computer Conf.*, June 1979, pp.313-317.

[11] Antonopoulos A M. Mastering Bitcoin: Unlocking Digital Crypto-Currencies. O'Reilly Media, 2014.

[12] Rivest R, Shamir A, Adleman L M. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 1978, 26(2): 96-99.

[13] Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology*, Stern J (ed.), Springer, 1999, pp.223-238.

[14] Lindell Y, Pinkas B. Privacy preserving data mining. *Journal of Cryptology*, 2002, 15(3): 177-206.

[15] Gaetani E, Aniello L, Baldoni R, Lombardi F, Margheri A, Sassone V. Blockchain-based database to ensure data integrity in cloud computing environments. In *Proc. the 1st International Italian Conference on Cybersecurity*, January 2017, pp.146-155.

[16] Ali M, Nelson J, Shea R, Freedman R J. Blockstack: A global naming and storage system secured by blockchains. In *Proc. USENIX Annu. Technical Conf.* (*ACT*), June 2016, pp.181-194.

[17] Liang X P, Shetty S, Tosh D, Kamhoua C, Kwiat K, Njilla L. ProvChain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In *Proc. the 17th IEEE/ACM International Symposium on Cluster Cloud and Grid Computing* (*CCGRID*), May 2017, pp.468-477.

[18] Cai C J, Yuan X L, Wang C. Towards trustworthy and private keyword search in encrypted decentralized storage. In *Proc. IEEE International Conference on Communications* (*ICC*), May 2017.

[19] Miller A, Juels A, Shi E, Parno B, Katz J. Permacoin: Repurposing bitcoin work for data preservation. In *Proc. IEEE Symp. Security and Privacy* (*SP*), May 2014, pp.475-490.

[20] Frikken K B. Secure multiparty computation. In *Algorithms and Theory of Computation Handbook*, Atallah M J (ed.), Chapman and Hall/CRC, 2010.

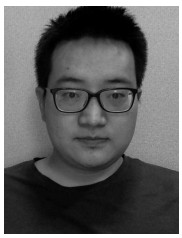
[21] Fouque P A, Poupard G, Stern J. Sharing decryption in the context of voting or lotteries. In *Financial Cryptography*, Frankel Y (ed.), Springer, 2000, pp.90-104.

[22] Baudron O, Fouque P A, Pointcheval D, Stern J, Poupard G. Practical multi-candidate election system. In *Proc. the 20th Annu. ACM Symp. Principles of Distributed Computing* (*PODC*), August 2001, pp.274-283.

Bao-Kun Zheng received his M.S. degree in computer science from the School of Information, Renmin University of China, Beijing, in 2010. He is currently a Ph.D. candidate at the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, and an associate professor at the School of Information Management for Law, China University of Political Science and Law, Beijing. His research interests include blockchain, network and information security.

Lie-Huang Zhu received his Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, in 2004. He is currently a professor at the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. His research interests include security protocol analysis and design, group key exchange protocols, wireless sensor networks, and cloud computing.

Meng Shen received his B.Eng. degree in computer science from Shandong University, Jinan, in 2009, and his Ph.D. degree in computer science from Tsinghua University, Beijing, in 2014. He is currently an assistant professor at the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. His research interests include privacy protection of cloud-based services, network virtualization, and traffic engineering. He was a recipient of the Best Paper Runner-Up Award at IEEE IPCCC 2014.



Feng Gao received his B.Eng. degree in computer science from the School of Software Engineering, Beijing Institute of Technology, Beijing, in 2010. He is currently a Ph.D. candidate at the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. His research interests include data privacy, blockchain and smart grid.

Chuan Zhang received his B.Eng. degree in network engineering from Dalian University of Technology, Dalian, in 2015. He is currently a Ph.D. candidate at the School of Computer Science and Technology, Beijing Institute of Technology. His research interests include secure data services in cloud computing, security and privacy in VANET, and big data security.

Yan-Dong Li received his B.Eng. degree and M.S. degree in computer science from the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, in 2015 and 2018, respectively. His research interests include blockchain, cloud computing security, and data privacy.

Jing Yang received her B.Eng. degree in computer science from School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, in 2012. She is currently a M.S. candidate at the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. Her research interests include security and privacy issues in Bitcoin.