Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Research on Data Security Protection Strategies in Cloud Computing Environment

Chengcheng Gao[1], Mingwei Liu[2], Ning Li[3], Xingda Gao[1], Xibao Wang[1], Shizhu Wu[4,✉]

[1] *The Network Security Lab, Yunding Technology Co., Ltd., Jinan, Shandong, 250000, China*
[2] *Information Technology Department, Shandong Jiuzhou Xintai Information Technology Co., Ltd., Jinan, Shandong, 250000, China*
[3] *Ministry of Information and Technology, Shandong Rural Credit Cooperatives Union, Jining, Shandong, 250000, China*
[4] *Information Technology Department, Shandong Sunshine Digital Technology Co., Ltd., Binzhou, Shandong, 256600, China*

ABSTRACT

In today's increasingly complex and dynamic network structure, cloud computing brings great convenience to computer users and meets people's requirements for rapid computer data processing. The article firstly analyzes the cloud computing architecture and the security threats existing in the cloud environment, and explains the importance of cloud computing access control mechanism to ensure data security, starting from the traditional access control. Then it introduces the multi-authority attribute access control scheme based on blockchain and elliptic curve improvement, on the basis of which it proposes a blockchain-based cloud data security sharing model and a blockchain transaction privacy protection scheme, which both meets the user data privacy protection needs and realizes privacy computing. Finally, the security of the two schemes is analyzed, and compared with other schemes with the same mechanism. The results show that the blockchain-based cloud data security sharing scheme has better performance and scalability, which shows a stable linear growth of 1x, and the time load introduced by this scheme while enhancing the security of the encrypted data sharing system is acceptable compared to the other schemes to satisfy the application scenarios with large-scale access requests. At the same time, the blockchain transaction privacy protection scheme ensures data privacy while the average time consumed meets the user's requirements for fast response.

Keywords: cloud computing, data security, blockchain, access control, privacy protection

1. Introduction

Cloud computing belongs to a new type of technology, which is developing rapidly in our country and has a wide range of applications. The use of cloud computing not only provides a platform for computer data processing, but also establishes a direction for the future development of computers. Along with

✉ Corresponding author.
E-mail address: 13365318522@163.com (S. Wu).
Received 15 February 2024; Revised 25 March 2024; Accepted 15 December 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-447

the continuous development of cloud computing, thus derived from the concept of cloud security, the user can use cloud computing to enhance the efficiency of data processing, to ensure data information security [1-4]. Affected by the openness of the Internet, computer data processing there are security risks, such as data leakage, illegal theft, etc., this part of the security problem directly affects the computer data processing security, but also threatens the personal and business user data security, so it is important to take appropriate response strategies to improve data security [5-8].

In cloud computing environment, data storage and processing is done through cloud service providers. This means that the user's data will be stored on the cloud servers and no longer on the user's own local device [9-10]. Such an architecture brings data security risks can be addressed by adopting privacy protection strategies including authentication, data encryption, regular backups, privacy policy review, data partitioning and access control, anonymization, and user education, etc. By adopting these strategies, users can better protect their data security and privacy while enjoying the convenience and benefits of cloud computing [11-14]. And with the further development of technology and the increasing demand of users for data security and privacy, data security and privacy protection strategies in cloud computing environment will receive more attention and improvement [15-16].

Literature [17] discusses the architecture, concepts and shortcomings of cloud computing based on the analysis of existing research results, and proposes a privacy protection framework. It also discusses the extension and layered key of KP-ABE, CP-ABE, etc., proposes the research challenges and development direction of privacy protection in cloud computing environment, and emphasizes the corresponding privacy protection laws, which aim to make up for the technical deficiencies. Literature [18] launched a systematic analysis of data security and privacy protection issues related to cloud computing, examining current solutions. It also provides an outlook on the research work on data security and privacy protection issues in cloud environments. Literature [19] provides a systematic introduction to cloud computing and points out that it brings with it issues related to data security, which are key barriers to better utilization of cloud computing services. By analyzing the basic issues of cloud computing, the issues of data security and privacy protection in cloud computing are discussed. Literature [20] describes the research progress on privacy security issues from the perspective of privacy security protection techniques in cloud computing. Some privacy security risks in cloud computing are introduced and a privacy security protection framework is proposed. The current challenges and possible future research directions are also discussed. Literature [21] emphasizes the important role of cloud computing and proposes a security-aware cloud model. It is emphasized that the use of this model not only improves the data security of the cloud but also the return on investment of the services provided by the cloud is improved. Literature [22] describes the security issues arising due to cloud computing and the recently proposed solutions. It also outlines the security vulnerabilities in mobile cloud computing and discusses the unresolved issues and future research directions. Literature [23] highlights the applications of cloud computing and the data security issues faced by the users and attempts to provide an overview of the relevant research in this area, with a review that categorizes the methods based on the type of method and the type of validation used to validate the method. Literature [24] shows the important role of data security and privacy protection in big data environment, reveals the shortcomings of current encryption methods, and proposes user-authorized data access management techniques based on attribute encryption and secret distribution techniques.

The study first focuses on analyzing the service system architecture and the current security threats in the cloud environment, and discusses the importance of access control to ensure information security. Then it illustrates the system framework and system model of the data sharing scheme, describes the whole data sharing process in detail through four stages: system initialization, data encryption, data access and data decryption, and analyzes the smart contract used. On this basis, the blockchain-based cloud data security sharing model and the blockchain transaction privacy protection scheme are proposed, and experiments are conducted to verify the security and efficiency of the

scheme, and the final test proves that the scheme in this paper can meet the demand for secure access to data within the scope of blockchain applications.

2. Data Security Analysis in Cloud Computing Environment

2.1. Cloud Computing Architecture and Security Threats

2.1.1. Cloud Computing Architecture. Cloud computing is not a new technology exactly, but a service model applicable to current big data processing that was spawned after the maturity of a variety of key technologies. Cloud computing centralizes the management of data resources, unifies scheduling and optimization, and handles large-scale data. For enterprises or individuals, the purpose of using the cloud is to rest assured that the data will be handed over to the cloud for management without worrying about information security issues such as illegal access to the data, reducing the cost and energy consumption of managing the data, and enjoying the maximum value of the data [25]. The user's request for data is unpredictable, and cloud computing supports the user's ability to dynamically adjust data resources and elastically provide resources to the outside world. Cloud computing architecture is shown in Figure 1, the core service layer provides services to meet the diverse needs of users.

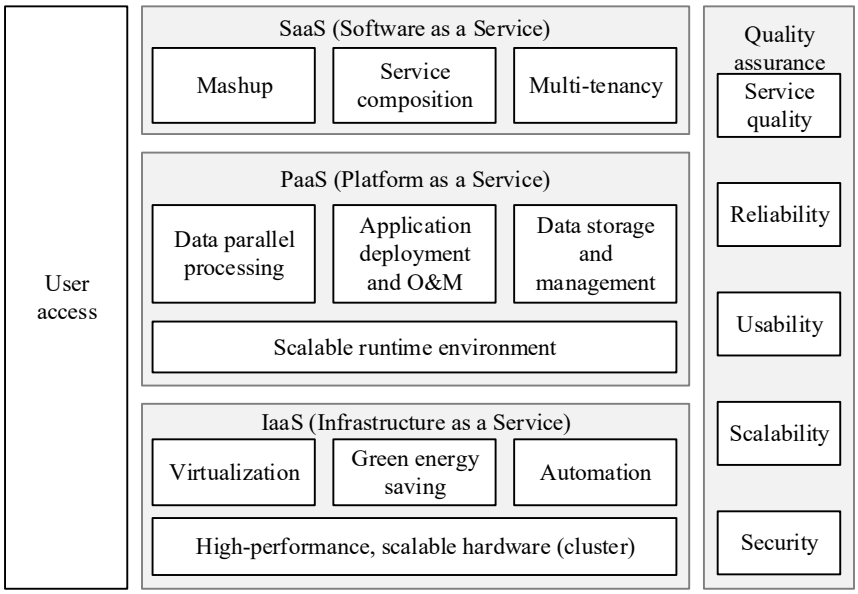

Fig. 1. Cloud computing architecture generation

2.1.2. Cloud Computing Security Threats. Cloud computing realizes the security management of the whole system by deploying data-centralized computing centers. The scale of data is getting bigger and bigger over time, coupled with the characteristics of cloud environment such as many users, complex data, and frequent user demands, the security risks endured are also increasing gradually, and it is more likely to be attacked by illegal users. The current risks in access control system are mainly the following four points:

1) Access rights risk. After the data is uploaded to the cloud service provider, due to the problem of distrust between people and the cloud, if the user's data permissions are leaked, it may lead to other users or the cloud service provider to snoop on the leaked data, programs, or the user's attributes are leaked, the data security is threatened, and the user's privacy is exposed.

2) Data transmission risk. Cloud for the public, the user can apply for access to the cloud data, data uploaded to the cloud service process, if the data confidentiality is not guaranteed, or data encryption,

but not enough protection of the cipher text, will lead to the transmission process by illegal users to steal, decrypting the cipher text to get the data in plain text, threatening the security.

3) Virtualization environment security risks. The cloud environment is a virtual resource pool with centralized data management, and the system will dynamically adjust the data according to the user's request. The current access control technology adopted in the cloud environment ignores the protection of ciphertext, which puts the data at risk of leakage. This, coupled with the lack of strong data isolation capabilities in the system architecture, can facilitate hacker attacks, taking advantage of the loopholes that exist in the imperfect access control mechanism and using sharing techniques to attack the data.

4) Data integrity is compromised. Data resources stored in a distributed cloud environment may also be illegally tampered with or destroyed, in the information age, data loss and destruction means loss of property, and safeguarding data integrity from destruction is also a very important aspect of cloud computing access control mechanisms.

2.2. Analysis of Cloud Computing Access Control Mechanisms

2.2.1. Authentication-based access control. When using identity-based cryptography for user authentication, the system has to establish a special channel to distribute private keys to different users to ensure security, the disadvantage is that if the private key center is attacked, it will directly affect the security performance of the user's identity information, and this kind of access control does not completely solve the data security problems existing in the current cloud environment, and it is unable to realize fine-grained access, and there is a lack of privilege revocation when the ciphertext is updated. There are still deficiencies in the revocation of privileges when the ciphertext is updated.

2.2.2. Attribute-based encryption access control. The access control mechanism based on attribute encryption is shown in Fig. 2. Users can download the encrypted data from the cloud service provider and decrypt it only by meeting the corresponding attribute conditions, which simplifies the operation, reduces the amount of computation, and is more flexible than the other schemes, and is suitable for cloud environments with a large number of users and a huge amount of data. Although CP-ABE scheme has been more suitable for cloud environment than other schemes, there are still some legacy problems that have not been solved, such as attribute revocation problem and key leakage problem.

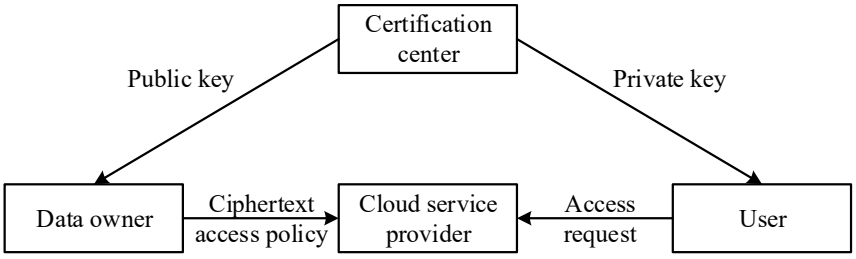

Fig. 2. Access control mechanism based on attribute encryption

2.3. Cloud Computing Data Encryption Technology

2.3.1. Symmetric encryption. Symmetric encryption technology uses the same key to encrypt and decrypt data operations, which is characterized by fast encryption speed and high efficiency. Applying symmetric encryption algorithms to cloud computing makes data encryption simpler and more convenient for users and cloud service providers, and the shorter key reduces the computational overhead, which is suitable for simple encryption in cloud computing to save computational overhead.

2.3.2. Asymmetric encryption techniques. The key used for encryption and the key used for decryption in asymmetric encryption algorithms are different, and the two keys can not be deduced from one of them to the other. The RSA algorithm is a commonly used algorithm in asymmetric encryption algorithms, the private key exists locally and the cipher text is transmitted, even if it is intercepted, it is not easy to be cracked in terms of the currently known data attacks, and theoretically, it is the most mature and perfect public key cryptosystem at present [26]. The high security of this encryption technique can guarantee data confidentiality and can be applied in cloud computing for key distribution.

The main steps of RSA algorithm are as follows:

1) Key generation:

(1) Choose prime number x, y (x, y confidential) and compute the open modulus r and Euler functions $\psi(r)$: $r = xy$ and $\psi(r) = (x-1)(y-1)$.

(2) Choose random number z, satisfying $1 < z < \psi(r)$, and generate parameters k; $k = z^{-1} \bmod(\psi(r))$.

(3) This yields the public key (z, r) and the private key is (k, r).

2) Encryption:

Encrypt the plaintext m to obtain the corresponding ciphertext c;

$$c = m^z \bmod r \tag{1}$$

3) Decryption:

Decrypt the ciphertext c to get the corresponding plaintext m: $m = c^k \bmod r$.

Where $r = xy$ denotes the length of the key, z satisfies $\gcd(z, \psi(r)) = 1$, and after calculation, the public key is (z, r) and the private key is (k, r).

3. Data Security Protection in Cloud Computing Environment

3.1. Blockchain and Elliptic Curve Based Access Controls

3.1.1. Overall architecture. The overall architecture of the IoT data sharing solution studied in this chapter consists of five layers: data collection layer, user layer, storage layer, interaction layer and access control layer.

3.1.2. System model. There are six entities in the system model of the research proposal: certificate authority (CA), IPFS, attribute authority, data owner, data visitor and blockchain network.

CA: CA serves as the entity of MSP in Fabric, which is mainly responsible for the registration of users and issuance of ID certificates in Fabric blockchain, and initializes the whole system by setting the security parameters and collecting the public keys of attribute authorities.

IPFS: IPFS, as the main container for storing data in the system, provides off-chain data storage service, which is used to solve the problem of insufficient on-chain storage capacity, i.e., to realize off-chain storage and on-chain retrieval.

Data Owner (DO): DO is the owner of data resources, which protects the data by two encryption algorithms. The encrypted ciphertext after attribute encryption is used to construct the Data Element Table (DET) and smart contracts are invoked to upload the DET.

Data Visitor (DV): the DV is a user who wishes to access and use the DO data. The DV requests the decryption key associated with its identity attributes through the blockchain. If the DV's attributes satisfy the access policy of the ciphertext, the data can be successfully accessed and decrypted.

Fabric Blockchain Network: Fabric, as the underlying platform of the system solution, is mainly responsible for storing relevant data information, accomplishing data sharing operations through

relevant smart contracts, and providing a transparent and auditable ledger for on-chain transactions. The data sharing system model studied in this chapter is shown in Figure 3.

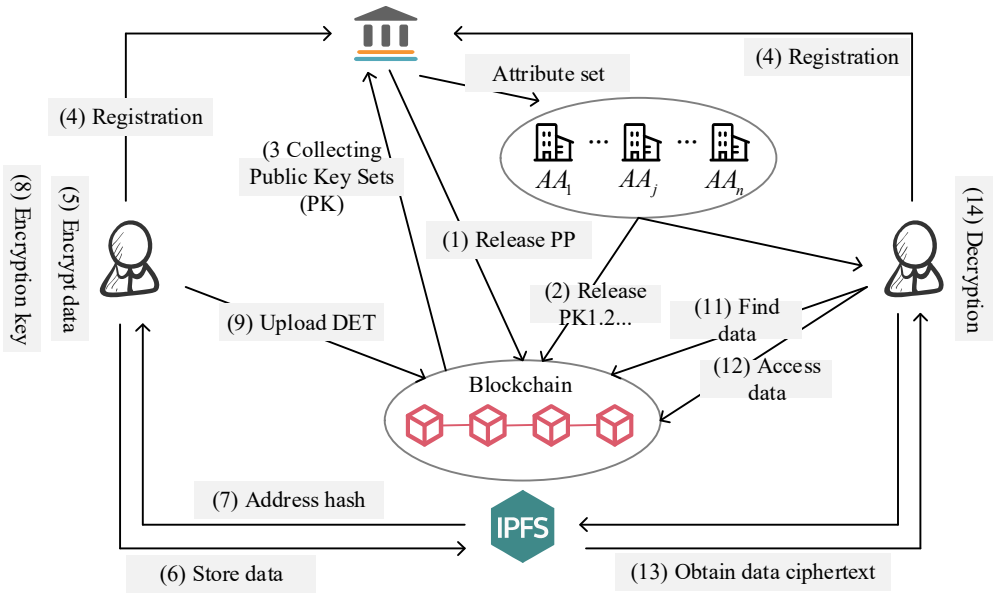

Fig. 3. System model

3.1.3. Data-sharing processes:

1) System initialization. CA runs the system initialization algorithm $System\_Setup(q \to PP)$, inputs a security parameter q, gets the global parameters of the system PP, and publishes them into the blockchain.

The details of the algorithm are as follows: the system initialization algorithm takes as input a large prime number q as a safety parameter and outputs a common parameter $(q, GF(q), E, G, r, h, H)$ of the system. $GF(q)$ is a finite field of order q and E is an elliptic curve in that finite field, where GR is a cyclic subgroup of order r containing all the points E on the elliptic curve. A point in GR is chosen as the generator G and any point in the cyclic subgroup GR can be generated by multiplying G with an integer i, where $i < r$. The positive integer h is an auxiliary factor mainly used in the computation of $h \cdot r = |E|$. For each user in the system there is a unique identifier GID and a hash function $H: \{0,1\}^* \to \square_r^*$ is chosen to map the user's identifier GID to an element of the Z_r.

2) Data Encryption. This stage mainly involves DO uploading data related information by mixing encryption and then invoking the smart contract in the blockchain.

DO construct the access policy (A, ρ) corresponding to this data and execute the encryption algorithm $Encrypt\{(PP, (A, \rho), Key, \{PK\}) \to CT\}$ to encrypt the AES encryption key Key to get the key ciphertext CT. The details of the algorithm are as follows:

(1) First the key Key to be encrypted has to be mapped to a point M on the elliptic curve E, then an integer $s \in Z_r$ is chosen at random and $C_0 = M + sG$ is computed.

(2) The encryption algorithm takes as input the access policy set by the DO, and then outputs the access matrix A corresponding to $n \times l$ for this access policy, and maps the attributes in the access policy to the matrix A by function ρ.

(3) Two random column vectors $v=(s,v_2,\ldots,v_m)\in Z_r$ and $u=(0,u_2,\ldots,u_m)\in Z_r$ are selected and $\lambda_x=A_x\cdot v$ and $\omega_x=A_x\cdot u$ are computed respectively, where A_x represents the x th row of the matrix.

(4) Finally, the ciphertexts $C_1,x=\lambda_x\cdot G+\omega_x\cdot PK_{\rho(x)}$ and $C_2,x=\omega_x\cdot G$ are computed.

3) Data Access. This phase focuses on the DV applying for the attribute sub-token and accessing the DO's uploaded data element table by invoking the smart contract.

DV applies for attribute identity through blockchain. All attribute authority AAs in the system run Algorithm $KeyGen\{(PP,i,SK,GID)\rightarrow SK_{i,GID}\}$ based on DV's identity information to issue attribute sub-token $SK_{i,j,VID}$ for it. Details of the algorithm are as follows:

The attribute authority generates attribute key $SK_{i,VID}$ for attribute A_i for the user who has VID. The calculation is as follows.

$$S\,K_{i,VID}=a_i+H(VID)\cdot n \tag{2}$$

DV calls the smart contract ReadData to query the information in the data description part of the table of all data elements in the current blockchain ledger. Then it looks up the data it needs through the returned data description information, where DataID is the basis for subsequent querying of data.

4) Data Decryption. In order to successfully decrypt the ciphertext, DV first finds a suitable set of rows A_x in matrix A, where $(1,0,\ldots,0)$ needs to be among these rows, submits its identifier VID and $(C_2,x,\rho(x))$ for each row x. And then performs the computation.

$$\begin{aligned}&\sum(C_{2,x}\cdot S\,K_{\rho(x),VID})\\&=\sum(\omega_xG\cdot(a_{\rho(x)}+H(VID)n))\\&=\sum(\omega_xa_{\rho(x)}G+\omega_xH(VID)nG)\end{aligned} \tag{3}$$

Based on the calculations above, the calculations will continue.

$$\begin{aligned}&\sum C_{1,x}-\sum(C_{2,x}\cdot S\,K_{p(x),VID})\\&=\sum(\lambda_x\cdot G+\omega_x\cdot PK_{\rho(x)})-\sum(\omega_xa_{\rho(x)}G+\omega_xH(VID)nG)\\&=\sum(\lambda_x\cdot G-\omega_xH(VID)nG)\end{aligned} \tag{4}$$

Then DV randomly chooses an integer $c_x\in Z_r$ such that $\sum c_xA_x=(1,0,\ldots,0)$, and computes:

$$\sum c_x(\lambda_xG-\omega_xH(VID)nG)=sG \tag{5}$$

When $v\cdot(1,0,\ldots,0)=s$ and $u\cdot(1,0,\ldots,0)=0$, DV can be successfully decrypted out, plaintext Ma:

$$C_0-sG=M \tag{6}$$

3.1.4. Smart Contract Analysis. Smart contract analysis is divided into three parts: data upload contract, data lookup contract and data access contract.

1) Data upload contract: Data upload contract mainly defines some rules and processes for DO to upload data to the blockchain.DO uses data upload contract to upload DET to the blockchain. In order to reduce the storage pressure of the blockchain, DO uses the information of the data to build the DET, and the DataID is used as the retrieval key of the DET on the blockchain.

2) Data Lookup Contract: The Data Lookup Contract is to show the user part of the information of all DETs in the current blockchain, and DV calls the contract to search for the required data.

3) Data Access Contract: DV accesses the data based on the DataID and the attribute tokens it owns by invoking the Data Access Contract.

3.2. Blockchain-based secure sharing of cloud data

3.2.1. System components. In this paper, we propose a blockchain-based cloud data integrity verification scheme that is more efficient and secure than existing schemes. The uniqueness of our scheme is that instead of relying on centralized and costly Third Party Auditors (TPAs), blockchain technology is utilized to enhance data security and privacy and reduce computational overhead and communication overhead. In this system model, two entities are included, the data owner (DO) and the cloud service provider (CSP). Unlike the traditional scheme, this paper improves the security and reliability of the data integrity verification process by introducing blockchain technology into the system and completing the checking of data integrity with the help of Merkle hash tree, which is a data structure. And reduce the network resource overhead.

3.2.2. Data storage structure. Data immutability and consistency in blockchain is due to the consensus of nodes across the network, each node saves the historical information of the entire network, which ensures the integrity of the data. However, this also brings certain problems, with the rapid growth of nodes, the data also grows explosively. So the idea of storing the cloud data on the blockchain is not realistic, therefore, the data storage structure proposed in this paper is by storing the metadata generated from the source data and the logs of the cloud server's operations on the data as the metadata of the data on the chain through the blockchain network.

3.2.3. Hybrid encryption mechanisms. In hybrid encryption mechanism, the symmetric encryption algorithm AES is used to encrypt the data while the asymmetric encryption algorithm RSA is used to protect the symmetric key for secure transmission.

There are major steps in generating the parameters of RSA public key encryption scheme:

1) Choose two large prime numbers (usually called p and q): choose two sufficiently large prime numbers p and q. These two prime numbers should be chosen at random and have similar bit lengths.

2) Calculate N: i.e., the product of the two prime numbers, i.e.,:

$$N = p * q \tag{7}$$

N will be used as the modulus in the RSA algorithm.

3) Calculation $\phi(N)$: Calculate the Euler function $\phi(N)$, where

$$\phi(N) = (p-1)(q-1) \tag{8}$$

$\phi(N)$ denotes the number of positive integers less than n and mutually prime with n.

4) Choose public key e: Choose an integer e such that $1 < e < \phi(N)$, and e are mutually prime with $\phi(N)$. A common choice is to choose a small prime number, such as 65537 (a common value).

5) Compute private key d: Compute private key d such that d is the multiplicative inverse of e with respect to modulo $\phi(N)$, i.e., satisfies the equation:

$$d * e \equiv 1 \bmod \phi(N) \tag{9}$$

In order to prevent malicious nodes from obtaining useful information by analyzing the hash tags, RSA based probabilistic encryption algorithm is used to protect the privacy of the data. The algorithm implementation process is as follows:

1) Generate key

Generate public key $pk = (e, n)$ and private key $sk = (d, n)$ in the local system

2) Encryption process

Equalize the plaintext m into the data segment $(m_1, m_2, ..., m_n)$ of length $L(L < \log_2 n)$; and generate a random number $r(r > 0)$, the length of r is the same as the data segment; and then encrypt the data segment. The specific realization algorithm is: the data segment m_i and r modulo multiplication, to obtain the ciphertext segment $(c_1, c_2, ..., c_j)$, using the public key e encryption to get the decryption key μ.

$$\mu = Enc(r, e) = r^e \bmod n \tag{10}$$

$$c_j = (m_i r) \bmod n, i = 1, 2, ..., j \tag{11}$$

Combine $\mu, c_1, c_2, ..., c_j$ in series to become ciphertext c.

3) Decryption process

Before decryption, the receiver needs to split the ciphertext to reduce $\mu, c_1, c_2, ..., c_j$; then use the private key d to decrypt to get $r = Dec(\mu, d)$; then use r to decrypt to get the message segment m_i;

$$r = Dec(\mu, d) = \mu^d \bmod n = r^{ò * d} \bmod n \tag{12}$$

$$m_i = Dec(c_i, r) = (c_i r^{-1}) \bmod n = m^i \, r r^{-1} \bmod n, i = 1, 2, ..., j \tag{13}$$

Finally, the message segments $(m_1, m_2, ..., m_j)$ are concatenated to obtain the plaintext m. In order for r to exist in the case of an inverse r^{-1}, r should be chosen to be mutually prime with n.

3.2.4. Blockchain-based cloud data security sharing model:

1) Authentication Management. Data transmission itself faces many problems, such as integrity, privacy and reliability. In order to ensure the security and trustworthiness of data, the primary consideration is the effective management of user identity.

2) Data Integrity Verification Protocol. First, DO generates the key pair corresponding to the encrypted file block and its hash tag, then sends a verification request to the blockchain network, and at the same time sends the public key pk along with it, since all nodes in the blockchain network are anonymous, only the node of the file block that matches with the public key can process the request. When the node receives the verification request, it looks up the latest hash tag of the file block in the blockchain and constructs a Merkle hash tree to return that proof.

Next, the DO (data owner) initiates a challenge Chal to the Cloud Service Provider (CSP), which constructs a Merkle hash tree for the file block through hash operations and returns that proof to the DO (data owner), at which point the DO (data owner) compares the two proofs for consistency, and if they are consistent, then the data is complete.

3.3. Blockchain-based transaction data privacy protection

3.3.1. Transaction Privacy Protection Program System Model. The system model of the blockchain transaction privacy protection scheme is shown in Figure 4. The scheme involves three entities: the payer, the payee, and the cloud server.

The payer encrypts the transaction amount and uploads it to the cloud server, and also has the power to decrypt and query the transaction result.

The cloud server stores the transaction amount encrypted by the payer, creates the payer's blockchain transaction and broadcasts it to the blockchain network according to the payer's transaction requirements. In this scenario, the cloud server is honest and the cloud server will perform the transaction calculation based on the payer's transaction and return the correct result to the payer.

Since the transaction amount is encrypted, the cloud server cannot know the exact transaction amount of the payer and can only help the payer to perform the transaction calculation and create the transaction. In this solution, the cloud server verifies the legitimacy of the transaction without disclosing the payer's privacy. The cloud server packages and broadcasts the legal transactions and rejects the packages for illegal transactions.

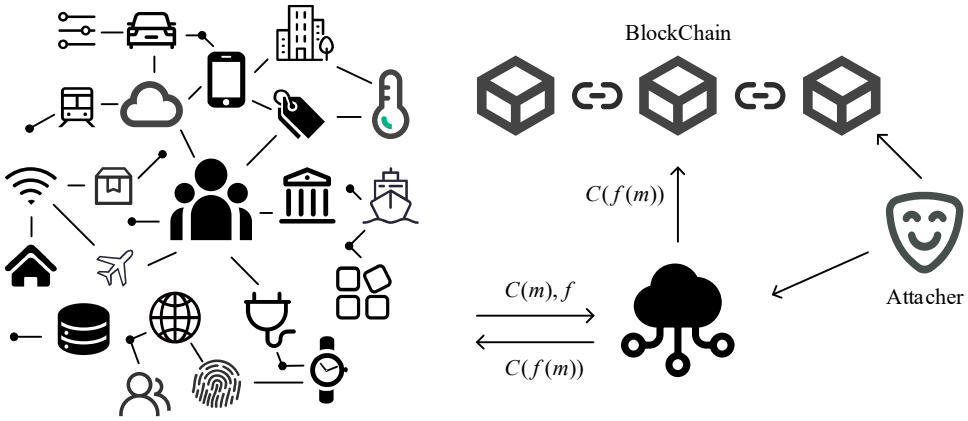

Fig. 4. System model of blockchain transaction privacy protection scheme

3.3.2. System implementation program:

1) Initialization stage. This initialization phase can be executed in advance. In this phase, the payer Alice initializes its account, i.e., generates its own public-private key pair for the multi-key fully homomorphic encryption algorithm. This phase is divided into two parts:

(1) Alice selects security parameter λ and circuit depth d and runs the initialization algorithm to generate system parameter pp.

$$pp = MKFHE.Setup(1^{\lambda}, 1^{d}) \tag{14}$$

(2) Combining the key generation algorithm of threshold encryption with the key generation algorithm of multi-key fully homomorphic encryption, Alice and the cloud server get the public key pk of the multi-key fully homomorphic encryption algorithm and its respective share of private key $sk_i (i = 0,1)$.

2) Encrypted Amount Phase. Once the payer Alice determines the transfer amount m, Alice will run the MKFHE encryption algorithm to encrypt the amount m.

$$c(m) = MKFHE.Enc(pk, m) = (U, C) \tag{15}$$

Alice then sends $c(m)$ to the cloud server.

3) Extending Ciphertext Phase. The cloud server runs the algorithm $MKFHE.Expand$ to extend Alice's ciphertext and outputs ciphertext $C(m)$. This ciphertext can be manipulated by both Alice and Bob, and the nature of the homomorphic cipher allows the user to perform ciphertext computation on the transaction amount via the cloud server.

$$C(m) = MKFHE.Expand((pk_A, pk_B), c(m)) \tag{16}$$

In this way, the cloud server converts the single-key encrypted transaction amount ciphertext into a multi-key ciphertext, so that this homomorphic ciphertext is converted from a ciphertext that can only be performed by the single-key owner for homomorphic computation to a ciphertext that can be performed by all participants for homomorphic computation.

4) Transaction Verification Phase. Correctness verification of the transaction: The correctness verification of the transaction ensures that Alice has sufficient assets to execute the transaction. The cloud server performs a subtraction operation on Alice's current balance and the transfer amount to ensure that the result is greater than or equal to 0. The cloud server randomly selects a monotonic decreasing function $f(\cdot)$. The cloud server executes this monotonic decreasing function on the ciphertext $c_B(m_{0,B})$ that obtains the amount of the transaction, and outputs the result of the computation $c_B(f(m_{0,B}))$. The cloud server performs the same operation on the homomorphic ciphertext $C_A(m_{0,A})$ of Alice's balance, and obtains the result of the computation $C_A(f(m_{0,A}))$.

The cloud server executes the ciphertext expansion algorithm on the computed result ciphertext to obtain the expanded ciphertext $C_B(f(m_{0,B}))$. Subsequently, the cloud server utilizes the property of MKFHE and executes the algorithm $MKFHE.Eval$ to perform homomorphic subtraction operation on the two homomorphic ciphertexts obtained.

$$C_A(w) = C_B(f(m_{0,B})) - C_A(f(m_{0,A})) \tag{17}$$

After getting the calculation result, the cloud server executes the threshold decryption algorithm to get the decryption result w. The cloud server checks whether the result w is greater than or equal to 0. If the result is greater than or equal to 0, the transaction is legitimate; otherwise, the transaction is rejected.

5) Update Ledger Stage. After completing the verification of transaction legitimacy, the cloud server updates the public ledger. The cloud server performs the ciphertext expansion algorithm on the transaction amount cipher to get the expanded ciphertext $C_B(m_{0,B})$. Then, the cloud server performs the homomorphic subtraction operation:

$$C_A(m_{1,A}) = C_A(m_{0,A}) - C_B(m_{0,B}) \tag{18}$$

The result of the operation $C_A(m_{1,A})$ is the homomorphic extended ciphertext of the current balance of Alice, the payer, after completing the transaction. The cloud server directly utilizes the result of this operation to update Alice's ledger. For Bob, the payee, the cloud server uses the transaction amount extended ciphertext $C_B(m_{0,B})$ to update his ledger.

4. Performance evaluation and analysis

4.1. Data Sharing and Access Control Security Analysis

In this section, we analyze the typical threat attack models that the proposed scheme can defend against. For encrypted data sharing, it is essential to ensure that the confidentiality and integrity of the data are not compromised, and traditional proxy re-encryption mechanisms are often threatened by conspiracy attacks. Therefore, we focus on three threat attack models, namely, conspiracy attack, man-in-the-middle attack, and data tampering, to analyze the security.

4.1.1. Effectiveness analysis. In this section, the efficacy of the blockchain-based cloud data security sharing model is comprehensively evaluated, and the performance of the system is analyzed through several sets of designed experiments in terms of time consumption at each stage of data sharing, re-encryption key parameter testing, and scalability testing, respectively.

In order to evaluate the time cost of introducing the threshold proxy re-encryption mechanism in blockchain networks, several measurements of the time load of each stage of data sharing in the blockchain-based cloud data security sharing model were conducted.The threshold proxy re-encryption data sharing mechanism can be divided into five stages, which are: key generation, data encryption, re-encryption key generation, data re-encryption and data decryption. The starting point

of the time overhead histogram of each phase is set to 0. The height of the histogram of each phase represents the height of the time overhead, and the phase with lower time overhead will be stacked on the outside of the histogram with higher time overhead.A comparison of the time overhead of each phase of encrypted data sharing is shown in Figure 5. It can be found that the time overheads of each phase are key generation, data decryption, re-encryption key generation, data encryption and data re-encryption in descending order; the data re-encryption phase spends significantly more time than the rest of the phases, with an average of 143 ms. The rest of the phases are completed at the user's end, and the average time overhead is less than 31 ms, which is an acceptable latency for the user.

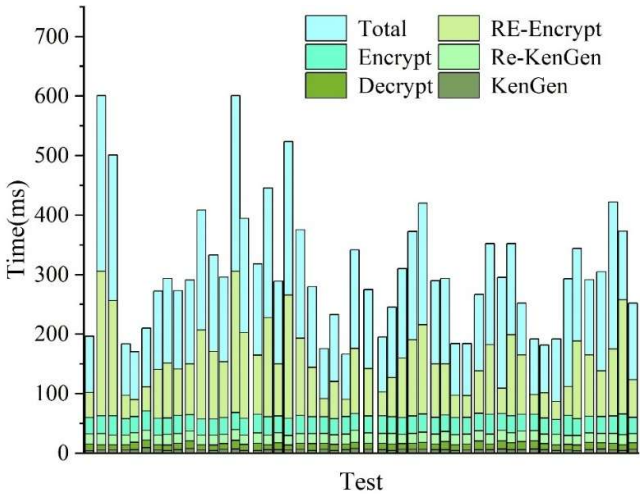

Fig. 5. The time overhead of each stage

In order to test the influence of the re-encryption key parameters of the blockchain-based cloud data security sharing model on the efficiency of data re-encryption processing, we set the re-encryption key parameters accordingly in order to observe the variation of the ciphertext conversion elapsed time in the data re-encryption phase, and the results are shown in Fig. 6.

The experimental results show that the more the split share of the re-encryption key, the longer the time spent in the data re-encryption phase. In addition, the time cost of the encryption process increases slightly as the threshold increases. This result shows that the efficiency of the threshold-based proxy re-encryption consensus mechanism is affected by these two parameters. The more the number of consensus nodes acting as proxy servers and the more re-encryption fragments required to combine new ciphertexts, the longer the consensus confirmation time will be, i.e., the longer it takes to complete the data re-encryption phase. However, the reliability and security of the data sharing system is also improved.

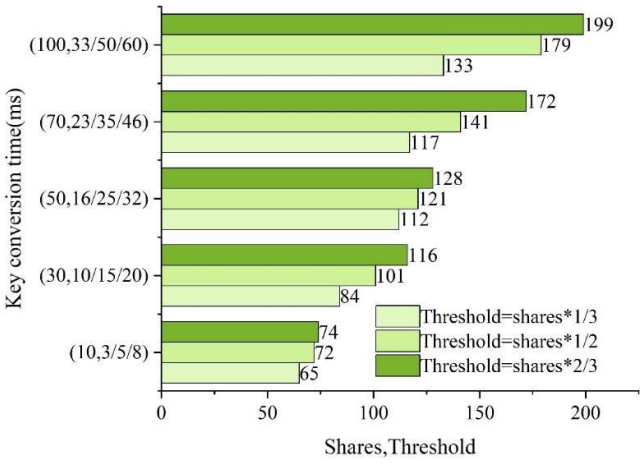

Fig. 6. Influences the results

On the other hand, the scalability of this scheme is tested by executing concurrent access requests from multiple data users. In the blockchain simulation experiment designed in the article, the time delay of the whole process of data sharing was recorded under a certain number of request scenarios by adding 5 requests each time and repeating the process until the total number of requests reaches 50, and each scenario was repeated 10 times and the average value was taken.

Fig. 7 shows the scalability test of the blockchain-based cloud data security sharing model system, the test results show that along with the increase in the number of access requests, the system latency shows a steady linear increase of 1x. The simulation results provide directions for system optimization, which can be used for subsequent improvements and increase efficiency.

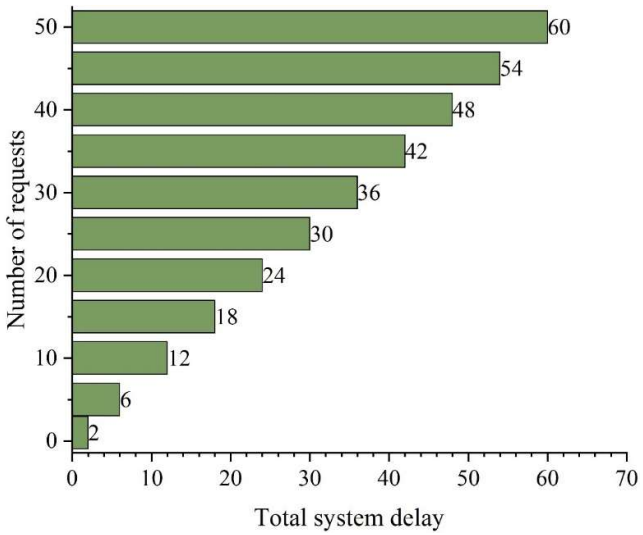

Fig. 7. System Scalability Test

4.1.2. Comparative analysis of programs. In this section, we mainly compare and analyze the functionality and performance of the proposed blockchain-based cloud data security sharing model with the schemes under the same mechanism. Table 1 shows the results of the functionality comparison, where "◎" denotes good functionality and "●" denotes average functionality compared to most schemes. This scheme combines threshold proxy re-encryption with blockchain with decentralized features. The scheme replaces a single semi-trusted proxy server with multiple consensus nodes in the blockchain network, which reduces the workload of the proxy server and improves the reliability of the system. In the mechanism of, the data owner is only responsible for the generation of the re-encryption key and bears a lower workload. In addition, the data owner can track and audit the whole process of data sharing and data re-encryption in the blockchain, which can effectively prevent the evil behavior of malicious nodes.

Table 1. Functional comparison of different schemes of agent reencryption mechanism

	Data confidentiality	Decentration	Multiple agents	Split the heavy encryption key	User load	Agent load	Auditability
Solution 1	◎	●	●	●	Middle	High	●
Solution 2	◎	●	●	●	High	Middle	●
Solution 3	◎	◎	●	●	High	Low	◎
Solution 4	◎	●	●	◎	Low	High	●
Solution 5	◎	●	●	●	Middle	High	●
Solution 6	◎	◎	●	●	Low	High	◎

Solution 1	◎	●	●	●	High	High	●
This article	◎	◎	◎	◎	Low	Low	◎

In terms of performance comparison, Figure 8 shows the results of the time overhead comparison for each phase of data sharing for the various schemes. This measurement is taken from the average of the CPU computation time for 100 executions of each operation type. In the data encryption and data decryption phases, our scheme requires slightly lower time overhead compared to other schemes such as Scheme 2, Scheme 5, and Scheme 6. Scheme 7 clearly has a higher time consumption because it performs the re-encryption process for a group of users using the broadcast encryption method. In the data re-encryption phase, the time delay of the proposed scheme in this paper increases significantly with an increase value of 82, but is still much lower than scheme 7 because the threshold proxy re-encryption and consensus algorithms require higher computational overhead.

The experimental results show that the proposed blockchain-based cloud data security sharing model is feasible and effective, and can improve the security and reliability of encrypted data sharing without introducing significant additional overhead and delay. Compared with the rest of the schemes, our system has unique advantages and can effectively adapt to the application scenario where data is stored in an untrustworthy distributed storage environment.

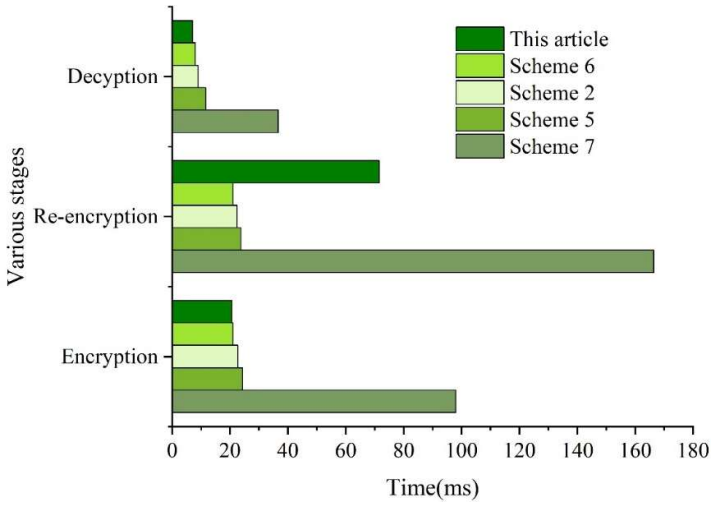

Fig. 8. Compares the time overhead of the shared stages

4.2. Transaction data security analysis

In order to evaluate the practical effect of the system model of the blockchain transaction privacy protection scheme, this section is based on the BGV fully homomorphic encryption algorithm for experimental analysis. The programming language used for the experiments in this chapter is Python, and the open source libraries include SEAL, PySEAL, and pyUmbral.

4.2.1. Parameter setting. The parameter settings for the experimental parameters in this chapter focus on the multimodal dimension and the number of multiplications (x^2 , x^4 , x^8). The example in the experiment is in the form of one bit array and the size of the array is $\frac{poly\_\bmod ulus\deg ree}{2}$. The specific parameter settings for the comparison experiment are as follows:

1) Instantiation of CCFT scheme based on BGV fully homomorphic encryption algorithm, in this section, the multimodulus dimensions are set to 2 × 1024, 4 × 1024, 8 × 1024, 16 × 1024, 32 × 1024. the multimodulus dimensions are varied according to the two-fold growth, and the average elapsed time for key generation, encoding, encryption, squaring, decryption, and decoding are evaluated.

2) Set the multimodal dimension to 32 × 1024, evaluate the average time spent for x^2, x^4 ,x^8, respectively, and observe whether the multiplier increases exponentially as its average time spent increases exponentially.

4.2.2. Experimental results. In this subsection, the experimental analysis of the blockchain transaction privacy protection scheme system model is conducted.The experimental analysis of the BGV scheme is carried out for the multimodal dimension and the number of multiplications to observe the average time consuming evaluation of the BGV varying with the multimodal dimension. For the evaluation experiments of the BGV scheme combined with proxy re-encryption, experimental tests are conducted for proxy re-encryption of ciphertext results in the system model of the blockchain transaction privacy protection scheme. Finally, the time performance of the blockchain transaction privacy protection scheme system model is evaluated experimentally with different number of threads. The time performance of the experiments in this chapter also follows the 2-5-10 response principle.

The average time consumption trend of BGV key generation with the dimension of multimodal number is shown in Fig. 9. From the figure, it can be seen that when the multimodulus dimension is at [2×1024,4×1024,8×1024], the variation of the average elapsed time for key generation fluctuates less, and the overall average elapsed time still varies within 90ms. When the multimodulus dimension is set to 16×1024, the average elapsed time reaches 194.28ms, compared with the first three parameters, the value of multimodulus increases by one order of magnitude, and the average elapsed time also increases by one order of magnitude. When the multimodal dimension is set to 32*1024, the average time consumed reaches 994.2ms, which still meets the user's requirement of fast response.

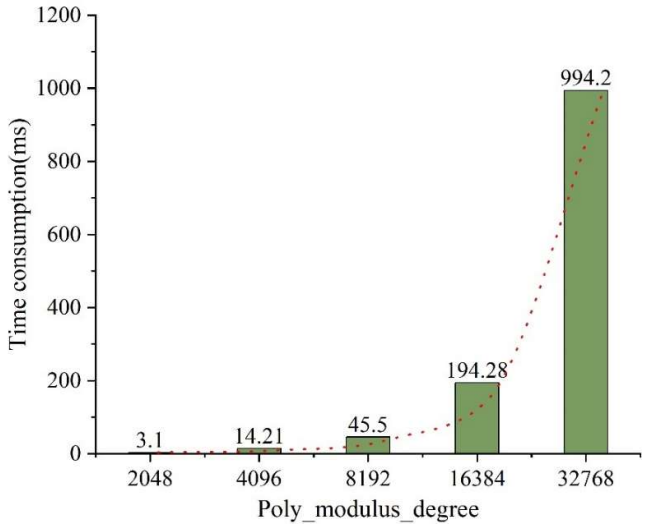

Fig. 9. Mean time consuming trend

The trend of the average elapsed time of the encryption operations of the BGV fully homomorphic encryption scheme as a function of the multimodal dimension is shown in Fig. 10. Overall, the fluctuation range of the average elapsed time of the encryption operation is small. When the multimodulus dimension is set to 32*1024, the average time consumed reaches 134.57ms, which is in the user-unperceived time range.

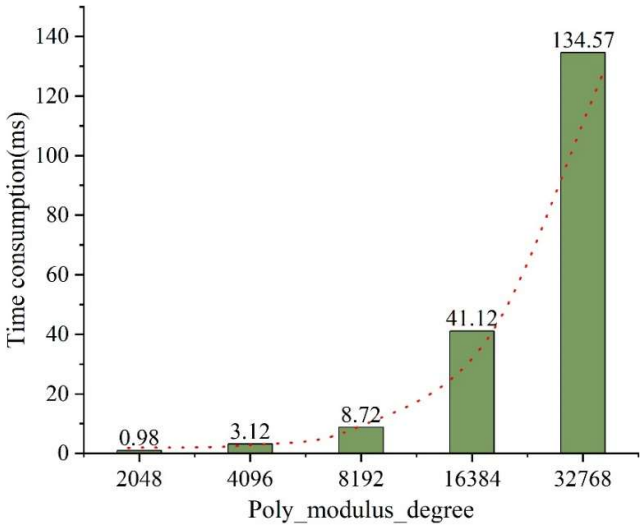

Fig. 10. Mean time consuming trend

Fig. 11 shows the average time consumption of the BGV fully homomorphic encryption square number finding operation with the change of multimodulus dimension. From the figure, we can see that when the multimodulus is set to $32 * 1024$, the average time consumed to find x^2 is 15.69ms, which is imperceptible to the user and meets the user's time requirement for fast response.

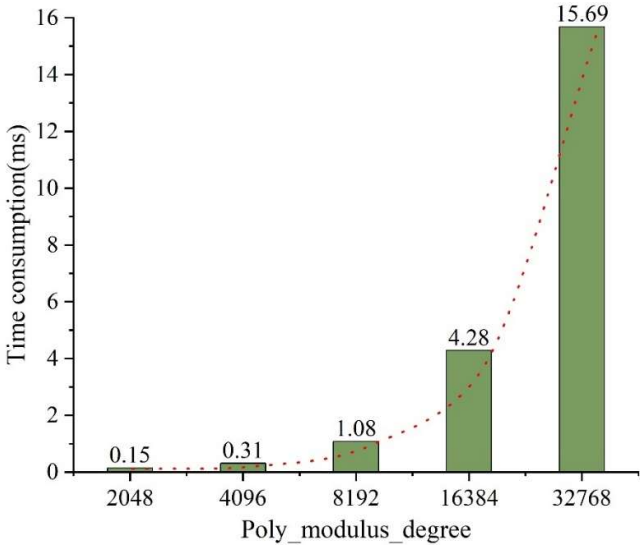

Fig. 11. Mean time consuming trend

Fig. 12 shows the trend of the average elapsed time of {x2, x4, x8} operations for a multimodule number of 32*1024. From the figure, it can be seen that the operations grow exponentially, but the growth of the average elapsed time is skewed towards parabolic growth. At computation x^8, the average elapsed time is 90.13ms.

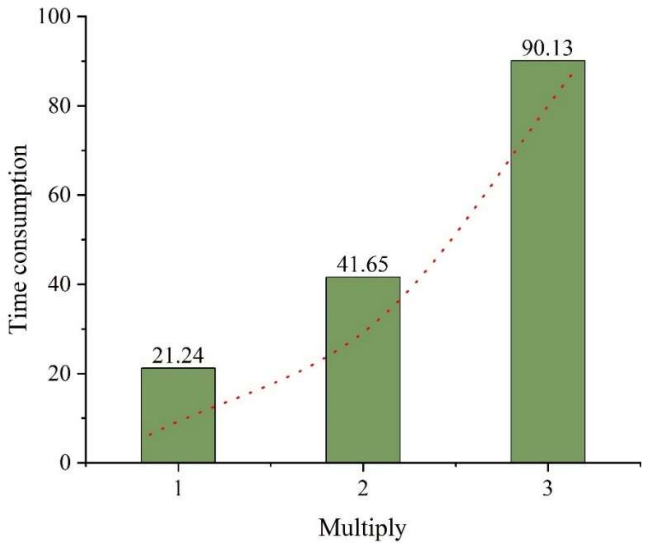

Fig. 12. Average time-consuming trend for different multiplication times

The trend of the average elapsed time of the decryption operation of the BGV fully homomorphic encryption with respect to the dimensionality of the number of multimodalities is shown in Fig. 13, where the average elapsed time increases with the increase of the number of multimodalities. From the figure, it can be seen that the average time consumption of the decryption operation is significantly less compared to the encryption operation under the same parameter conditions, and the time magnitudes are all within 45ms. When the multimodulus dimension is set to 32∗1024, the average time consumption is 41.31ms.

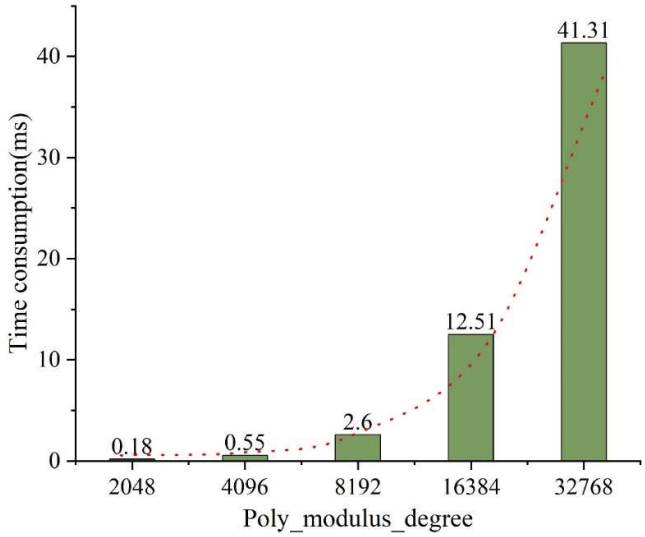

Fig. 13. Mean time consuming trend

Figure 14 shows the trend of execution time of all steps in the blockchain transaction privacy protection scheme with different number of threads. The multimodal dimension of this experiment is set to 32768 and the ciphertext volume is 8092.The average elapsed time of the system model of the blockchain transaction privacy protection scheme shows an overall decreasing trend with the increase of the number of threads, and the difference between the maximum and minimum average elapsed time is within 2s. When the number of threads is 8, the average elapsed time is 951.25ms, and the experimental results show that the blockchain transaction privacy protection scheme is able to satisfy the user's requirement for fast response under high security and large ciphertext volume. When the

number of threads is 16, it is affected by system resources, thread switching and other factors, resulting in worse time performance. Therefore, when deploying the blockchain transaction privacy protection scheme in practical applications, the number of threads should be selected according to the hardware configuration and performance requirements to achieve the fastest response time.

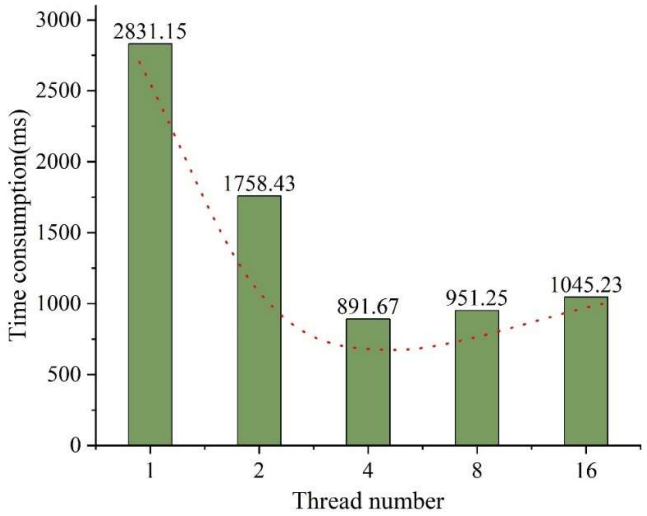

Fig. 14. Trends of the average execution time changes across the number of threads

5. Conclusion

The study proposes two schemes, a blockchain-based cloud data security sharing model and a blockchain transaction privacy protection scheme system model, around the cloud computing architecture to protect the data and privacy of users in the cloud computing environment. Finally, the security of the schemes is analyzed and the performance is evaluated. The experimental results show that the blockchain-based cloud data security sharing scheme has certain feasibility and effectiveness, and can improve the security and reliability of encrypted data sharing without introducing significant additional overhead and delay. At the same time, this scheme has unique advantages over the rest of the schemes, and the blockchain-based cloud data security sharing scheme can effectively adapt to the application scenario that the data is stored in an untrustworthy distributed storage environment. In addition, the average time consumption of the blockchain transaction privacy protection scheme is within 2s, which meets the response requirements of the client.

References

[1] Cao, Q., Schniederjans, D. G., & Schniederjans, M. (2017). Establishing the use of cloud computing in supply chain management. Operations Management Research, 10, 47-63.

[2] Sharma, P. K., Kaushik, P. S., Agarwal, P., Jain, P., Agarwal, S., & Dixit, K. (2017, October). Issues and challenges of data security in a cloud computing environment. In 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON) (pp. 560-566). IEEE.

[3] Alam, T. (2020). Cloud Computing and its role in the Information Technology. IAIC Transactions on Sustainable Digital Innovation (ITSDI), 1(2), 108-115.

[4] Varghese, B., & Buyya, R. (2018). Next generation cloud computing: New trends and research directions. Future Generation Computer Systems, 79, 849-861.

[5] Kumar, P. R., Raj, P. H., & Jelciana, P. (2018). Exploring data security issues and solutions in cloud computing. Procedia Computer Science, 125, 691-697.

[6] Kacha, L., & Zitouni, A. (2018). An overview on data security in cloud computing. Cybernetics Approaches in Intelligent Systems: Computational Methods in Systems and Software 2017, vol. 1, 250-261.

[7] Hajare, R., Hodage, R., Wangwad, O., Mali, Y., & Bagwan, F. (2021). Data Security in Cloud. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 8(3), 240-245.

[8] Ghorbel, A., Ghorbel, M., & Jmaiel, M. (2017). Privacy in cloud computing environments: a survey and research challenges. The Journal of Supercomputing, 73(6), 2763-2800.

[9] Markandey, A., Dhamdhere, P., & Gajmal, Y. (2018, September). Data access security in cloud computing: A review. In 2018 International Conference on Computing, Power and Communication Technologies (GUCON) (pp. 633-636). IEEE.

[10] Mushtaq, M. F., Akram, U., Khan, I., Khan, S. N., Shahzad, A., & Ullah, A. (2017). Cloud computing environment and security challenges: A review. International Journal of Advanced Computer Science and Applications, 8(10).

[11] Arafat, M. (2018, June). Information security management system challenges within a cloud computing environment. In Proceedings of the 2nd International Conference on Future Networks and Distributed Systems (pp. 1-6).

[12] Zhe, D., Qinghong, W., Naizheng, S., & Yuhan, Z. (2017, May). Study on data security policy based on cloud storage. In 2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids) (pp. 145-149). Ieee.

[13] Kaaniche, N., & Laurent, M. (2017). Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms. Computer Communications, 111, 120-141.

[14] Colombo, M., Asal, R., Hieu, Q. H., El-Moussa, F. A., Sajjad, A., & Dimitrakos, T. (2019, July). Data protection as a service in the multi-cloud environment. In 2019 IEEE 12th International Conference on Cloud Computing (CLOUD) (pp. 81-85). IEEE.

[15] Sauber, A. M., El-Kafrawy, P. M., Shawish, A. F., Amin, M. A., & Hagag, I. M. (2021). A new secure model for data protection over cloud computing. Computational Intelligence and Neuroscience, 2021(1), 8113253.

[16] Mishra, N., Sharma, T. K., Sharma, V., & Vimal, V. (2018). Secure framework for data security in cloud computing. In Soft Computing: Theories and Applications: Proceedings of SoCTA 2016, Volume 1 (pp. 61-71). Springer Singapore.

[17] Sun, P. J. (2019). Privacy protection and data security in cloud computing: a survey, challenges, and solutions. Ieee Access, 7, 147420-147452.

[18] Chen, D., & Zhao, H. (2012, March). Data security and privacy protection issues in cloud computing. In 2012 international conference on computer science and electronics engineering (Vol. 1, pp. 647-651). IEEE.

[19] Dinadayalan, P., Jegadeeswari, S., & Gnanambigai, D. (2014, February). Data security issues in cloud environment and solutions. In 2014 World Congress on Computing and Communication Technologies (pp. 88-91). IEEE.

[20] Sun, P. (2020). Security and privacy protection in cloud computing: Discussions and challenges. Journal of Network and Computer Applications, 160, 102642.

[21] Saeed, M. Y., & Khan, M. N. A. (2015). Data protection techniques for building trust in cloud computing. International Journal of Modern Education and Computer Science, 7(8), 38.

[22] Ali, M., Khan, S. U., & Vasilakos, A. V. (2015). Security in cloud computing: Opportunities and challenges. Information sciences, 305, 357-383.

[23] Soofi, A. A., Khan, M. I., & Amin, F. E. (2017). A review on data security in cloud computing. International Journal of Computer Applications, 96(2), 95-96.

[24] Kim, S. H., & Lee, I. Y. (Eds.). (2015). Study on user authority management for safe data protection in cloud computing environments. Symmetry, 7(1), 269-283.

[25] Pradeep Kanchan,V. Selvakumar,P. Lavanya,Ravi Kumar,Awakash Mishra,V. Haripriya & Gunveen Ahluwalia. (2024). Integration of IoT & cloud computing in mobile communication to breach limitation. International Journal of Information Technology(prepublish),1-7.

[26] Hadeel J. Shnaen & Sadiq A. Mehdi. (2023). Enhancing Key Exchange Security: Leveraging RSA Protocol in Encryption Algorithm Based on Hyperchaotic System. International Journal of Safety and Security Engineering(6).