Data Privacy Access Control Method Based on Ciphertext Policy Attribute-Based Encryption Algorithm

Chuangji Zhang[1(✉)], Weixuan Lin[2], and Yanli Zhang[3]

[1] Guangzhou Huali Science and Technology Vocational College, Guangzhou 511325, China
zhangcj21000@163.com
[2] Guangzhou Huashang Vocational College, Guangzhou 511325, China
[3] Metering Center of Jibei Power Grid Co. Ltd., Beijing 100045, China

Abstract. The data stored in the cloud contains a lot of privacy and confidentiality. The scalability of data privacy access control is weak, the attack resistance rate is low, and the risk rate of privacy disclosure is high. Therefore, a data privacy access control method based on ciphertext policy attribute-based encryption algorithm is proposed. The encryption scheme is designed based on the ciphertext policy attribute-based encryption algorithm as the formulated privacy policy. Aiming at the implementation and guarantee of privacy policy in cloud environment, a trusted execution method of privacy policy in cloud environment is proposed. Design a data access control model based on blockchain. The entities included in the model are data owner, data requester, data storage center, attribute authority and blockchain network. The test results show that the method has strong scalability, high fault tolerance rate, no third party, can achieve authorization, and the attack resistance rate is higher than 94%, and the risk rate of privacy leakage is low.

Keywords: Ciphertext Strategy · Attribute-Based Encryption Algorithm · Trusted Computing Technology · Data Privacy Access Control

1 Introduction

In recent years, with the large-scale application of cloud computing in real life, its own security issues have also attracted everyone's attention. Applications such as government departments and enterprise data outsourcing services have become an important part of cloud computing applications. However, because data is stored in a third party in cloud computing services and users access data through the network, the privacy access of confidential data has become a bottleneck restricting the development of cloud computing. In a sense, how to ensure the security of cloud computing services is the key to the sustainable and rapid development of cloud computing. The so-called cloud computing security involves three technical directions: data encryption, access control and security protection for virtual machines. In the information security system, access control is a very important part. Generally speaking, the role of access control in the information

B. Wang et al. (Eds.): ICMTEL 2023, LNICST 534, pp. 349–363, 2024.
https://doi.org/10.1007/978-3-031-50577-5_25

security system is to ensure that users or programs with rights can access the resources that have access, and at the same time to ensure that malicious or illegal users cannot access those protected resources. In the cloud environment, data storage and access are different from traditional methods. The resources in the cloud are dynamic and scalable, and the cloud service provider is not necessarily trusted. Therefore, how to securely implement the access control of data in the cloud is a work worthy of research.

Attribute-based encryption has attracted much attention because of its ability to achieve efficient one-to-many broadcast encryption and fine-grained access control. The basic attribute encryption technology is divided into ciphertext strategy and key strategy. Among them, the attribute-based encryption technology of ciphertext strategy is considered to be one of the most suitable technologies for data access control in cloud systems. This is primarily because it gives data owners more direct control over access policies and policy checks that take place "in the password". Therefore, how to provide a reliable escort for data sharing in the cloud computing environment, while protecting the privacy of users in all aspects, achieve safe, reliable and efficient access control is a work worthy of research. Based on the relevant research results of predecessors, the access control based on attribute-based encryption is undoubtedly very suitable for the cloud environment.

To sum up, cloud computing has occupied a certain position in people's work and life. Therefore, it is imperative to protect the security of cloud computing. Access control technology is an important method to ensure data security in the cloud. Therefore, the research on access control of cloud computing, especially the research on access control based on attribute encryption in cloud computing environment, will be of great significance in ensuring the security of cloud computing, and it is necessary to carry out related research in this area. Therefore, a data privacy access control method based on ciphertext policy attribute-based encryption algorithm is designed. The privacy policy is formulated according to the ciphertext policy attribute-based encryption algorithm, and the cloud privacy policy is credibly executed using the policy-driven idea. The data access control model based on the blockchain is constructed. The AES encryption algorithm is used to encrypt user data, and the CP-ABE is used to achieve privacy access.

2 Literature Review

Reference [1] combines hierarchical attribute encryption with linear secret sharing, and proposes a blockchain data privacy protection control scheme based on searchable attribute encryption, which solves the problem of privacy exposure in blockchain transactions.

Reference [2] proposes a blockchain-based cloud environment privacy protection access control framework. Using the account address of the node in the blockchain as the identity, redefine the access control authority of cloud data, and store it encrypted in the blockchain, this method can effectively protect the authorization privacy. However, the above methods still have the problems of weak scalability, low attack resistance rate and high risk rate of privacy leakage.

3 Data Privacy Access Control

3.1 Formulation of Privacy Policy

Design an encryption scheme based on the ciphertext policy attribute-based encryption algorithm as the formulated privacy policy [3]. Combining proxy re-encryption and ciphertext policy attribute to realize attribute revocation based on encryption technology, a CP-ABE-based attribute revocation encryption scheme is proposed in the multi-attribute authority environment. Using multiple attribute authorities reduces the security threat brought by a single attribute authority. The user's private key is jointly generated by the data owner and multiple attribute authorities, making it resistant to collusion attacks by multiple attribute authorities. The cloud storage server is responsible for storing ciphertext and performing proxy re-encryption operations, which reduces the workload of data owners and authorized agencies, and improves the efficiency of the entire system [4].

There are four main subjects in the system architecture of the CP-ABE scheme model of attribute revocation in the multi-attribute authority environment: data owner, data user, M attribute authority centers and cloud storage server.

(1) Data owner: The data owner customizes the access structure W for the data and generates random numbers for user decryption, and encrypts the data and uploads the ciphertext C_T to the cloud storage server [5].
(2) Cloud storage server: store the ciphertext C_T uploaded by D_O, and re-encrypt the ciphertext uploaded by D_O when the attribute authorization center AA_i notifies that the ciphertext needs to be updated.
(3) Attribute Authorization Center: There are M AA_i s in the system, which are mainly responsible for generating system parameters and master keys, managing different attribute sets of D_O, generating and distributing some private keys for users, and requiring AA_i to be fully trusted by users.
(4) User: The user of the data. When the version number verification is successful and the attribute satisfies the access structure corresponding to the ciphertext, the ciphertext can be decrypted [6].

Formal definition of the scheme: The scheme includes the following 8 algorithms:

(1) Setup: Multiple attribute authorization centers run the algorithm, first generate an attribute set U, each AA_i manages a different attribute set U, and generates system parameters and system master keys.
(2) KeyGen: This algorithm is jointly run by D_O and AA_i. Enter the attribute list L and the master key m_{sk} to generate the user private key component SK_L^i corresponding to the user attribute. The CP-ABE scheme model structure is shown in Fig. 1.
(3) Encrypt: This algorithm is performed by D_O. Input message m, specify access structure W, and output ciphertext C_T.
(4) Key Aggregation: This algorithm is executed by the user. After the user receives the private key component SK_L^i sent by the attribute authority, the user calculates the user's private key SK_L.
(5) Decrypt: After receiving the ciphertext, the data user first checks the version number of the private key, and performs decryption operation if it is consistent, otherwise

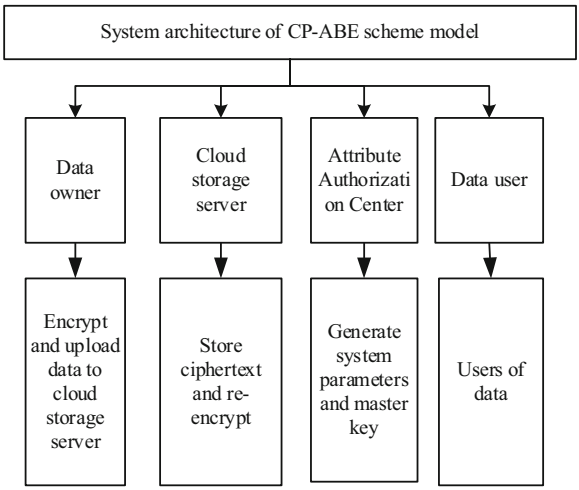

Fig. 1. System architecture of CP-ABE scheme model

updates the key and prepares to execute the ReEncrypt algorithm: the algorithm inputs SK_L and L, and outputs the plaintext message m.

(6) RekeyGen: When the user in the system needs to revoke, the attribute authorization center sends the new version number $u_{i,x}$ to the user through a secure channel, and outputs the proxy re-encryption key R_{sk}.

(7) ReEncrypt: This algorithm is performed by CSS. Input the ciphertext C_T and the re-encryption key R_{sk}, and output the re-encrypted ciphertext C'_T.

(8) Rekey: After receiving C'_T, the user checks whether the version number of his private key SK_L is consistent with the version number in C'_T. If they are inconsistent, output the same private key C'_T as before the attribute revocation; if they are consistent, output the updated key $SK'_L = SK_L$.

Scheme security model: In the security model, the adversary A fights B, and A can obtain the corresponding private key of the attribute set that satisfies its own access structure from B at the same time. The security model is constructed through the game of the adversary and the challenger. The model is to choose plaintext security, which consists of the following stages:

(1) Initialization

Adversary A selects an access policy to be challenged, as follows:

$$P = [P_1, P_2, \cdots, P_k] \tag{1}$$

In formula (1), P_k represents the k access sub-policy.
Send it to Challenger B.

(2) System establishment B runs the Setup algorithm and sends the generated system parameters to A.

(3) Query Phase 1

A submits its attribute set to B, and B runs the key generation algorithm to generate the corresponding private key and send it to A.

(4) Challenge stage

A submits two plaintexts M_α and M_β of the same length to B, and B runs the key generation algorithm to generate the corresponding private key and send it to A.

(5) Query Phase 2

A performs operations consistent with query phase 1.

(6) Guess

A guesses c if the following formula holds:

$$c' = c \tag{2}$$

In formula (2), c' represents the guessed c.
Then B wins the game. The advantages of B to play the game successfully are:

$$K(B) = \left| Q\left[c' = c\right] - \frac{1}{2} \right| \tag{3}$$

In formula (3), $Q\left[c' = c\right]$ means not equal to $c' = c$.
This completes the formulation of the privacy policy.

3.2 Trusted Implementation of Privacy Policy

3.2.1 Background of Trusted Execution

The trusted execution of cloud privacy policy is an important guarantee for user data privacy security. Aiming at the implementation and guarantee of privacy policy in cloud environment, a trusted execution method of privacy policy in cloud environment is proposed. The method adopts the policy-driven idea, aiming at the diversified privacy protection requirements of multi-tenant, and realizes the flexibility of privacy protection through distributed execution of privacy policy in the cloud environment. At the same time, combined with trusted computing technology, it prevents malicious modification of the cloud privacy policy by attackers, ensures the credibility of the implementation process of the privacy policy, and ultimately ensures the privacy and security of the user's stored cloud data.

The application scenarios of the trusted execution method of privacy policy in cloud environment are as follows: users can directly or indirectly interact with cloud service providers through proxy servers. After the proxy server receives the user request, it will distribute the user request to the distributed data processing server using methods such as load balancing. Thereafter, the processing server accesses the data storage server through the intranet to obtain the data content specified in the user request, forwards the data to the proxy server, and finally responds to the user's cloud resource request.

In view of the above scenarios, there is a risk of data privacy leakage in the cloud environment, and a trusted execution method of privacy policy in the cloud environment is

proposed. The method adopts a policy-driven method and combines trusted computing technology, which can effectively prevent attackers from maliciously tampering with cloud access control policies and ensure the privacy and security of user data.

Its architecture is as follows: distributed, each module involved in the implementation of the privacy policy is deployed on different independent nodes, and the data exchange between the modules is carried out through the intranet. These modules include: privacy policy enforcement module, privacy policy information module, privacy policy management module, privacy policy decision module, account storage module, container storage module, object storage module, policy repository storage module and data query module.

The trusted execution method of privacy policy in cloud environment mainly includes two parts: the distributed execution method of privacy policy and its implementation guarantee method [7]. The former is composed of various modules in the architecture to realize the distributed execution of the privacy policy; the latter ensures the trusted execution of the privacy policy by deploying a trusted execution module, a trusted storage module and a trusted communication module in the above architecture.

In the access control execution method in the cloud environment, there is still a lack of policy-driven privacy protection methods. However, the diverse and changing multi-tenant privacy protection requirements in cloud computing scenarios urgently require policy-driven privacy protection. In this regard, the designed policy distributed execution method adopts the policy-driven idea to study the distributed execution method of privacy policy in cloud environment. Based on the unified description of the privacy policy, the trusted execution method of the privacy policy can be deployed in a loosely coupled distributed manner on demand, which is more suitable for the diversity of tenant privacy protection requirements and the variability of privacy policies. It is more conducive to realizing the flexibility of privacy protection under the condition of a large cloud user group. The main implementation idea of the distributed execution method of privacy policy: After the cloud service receives the user access request, it searches for the metadata information of the user account, container and object according to the request to complete the user access request. At the same time, find a privacy policy that matches the user request, match the obtained privacy policy with the completed user request, and judge and respond to the user access request.

Now, the four main privacy policy execution function modules in the privacy policy distributed execution method, namely privacy policy enforcement module, privacy policy information module, privacy policy management module and privacy policy decision module, respectively, introduce the implementation of the method:

The privacy policy enforcement module is used to receive user requests, preprocess user requests in different formats, and convert them into a unified policy description language. The transformed user request is then forwarded to the privacy policy information module and the privacy policy management module distributed on different nodes. At the same time, receive the user request matching result sent by the privacy policy decision-making module, and judge whether to accept or refuse to respond to the user request according to the result;

The privacy policy information module searches the account, container, and object storage modules for the account, container, and object metadata corresponding to the user

request through the data query module. And use the attribute information in the metadata to complete the subject and object information in the user request, and the completed user request contains complete subject and object attribute information. Then, it is sent to the privacy policy decision-making modules distributed on different nodes in XML format based on the SOAP protocol, so that each module can judge the user's request;

The privacy policy management module uses the data query module to find the privacy policy in XML format that matches the subject and object information in the user request in the policy repository storage module, and obtains a policy subset that matches the subject and object information of the user request. The policy subset is composed of multiple matching policies in XML format, and the policy subset is sent to the privacy policy decision modules distributed on different nodes for it to judge user requests;

The privacy policy decision module is used to match the completed user request with the policy subset to judge whether the request is legal [8]. The matching method is to parse the attribute information of subject, object and operation in the user request, and match it with the attribute information of subject, object and operation of the policy in the policy subset one by one. Check whether there is a policy matching the subject and object in the user request in the policy subset, and return the result of whether there is a policy matching it to the privacy policy enforcement module. If the matching result is true, it indicates that the user request is valid, and if the matching result is false, it indicates that the user request is invalid.

There is still a lack of privacy policy implementation guarantee schemes in the distributed access control execution scheme in the cloud environment, which cannot be applied to the requirements of trusted implementation of privacy policies in the cloud environment. In the process of implementing the privacy policy, cloud service providers or malicious parties can tamper with the privacy policy, execute code, and hijack user requests. As a result, the privacy policy cannot be effectively implemented, forcing user data to be illegally obtained, and user privacy is violated, and it is difficult to detect. In view of the above problems, this section draws on the idea of trusted computing and proposes a guaranteed method for implementing privacy policy. The method ensures the credibility and integrity of the privacy policy and the code executed in each distributed node during the entire execution process through trusted storage, trusted execution, and trusted communication, and ensures the trusted implementation of the privacy policy.

3.2.2 Composition of Trusted Modules

The privacy policy implementation guarantee method is composed of three trusted modules, a trusted support module, a trusted storage module, and a trusted communication module, so as to ensure the trusted implementation of the distributed execution method of the privacy policy.

The trusted support module and the modules in the distributed execution method of the privacy policy are deployed on the same node. Taking the privacy policy enforcement module as an example, the trusted support module conducts initial trust measurement and real-time trust measurement for the privacy policy enforcement module by building a trust chain. The trusted chain is measured layer by layer from bottom to top, followed by the trusted platform module TPM that comes with the node chip, the core trust root

CRTM, the operating system bootloader, the operating system and the privacy policy enforcement module. The integrity of the code of each layer is guaranteed by the lower layer. Whenever the lower layer code completes the measurement of the upper layer code, it will be recorded in the measurement log. After that, the trusted platform module TPM located on the node chip invokes the underlying algorithm to calculate the summary value of the metric log. At the initial confidence measurement, this digest value will be stored in a Program Control Register (PCR) located within the TPM. In order to achieve the purpose that the data content cannot be tampered with, it is used to compare with the abstract value generated by the real-time credibility measurement, as the basis for verifying the real-time credibility measurement. If the PCR value generated by the real-time measurement is the same as the initial measurement PCR value, it is considered that the current state of the node is the same as that at startup, the code of each layer has not been tampered with, and the node is credible; otherwise, the node is considered untrustworthy;

The trusted storage module is deployed on the same node as the account storage module, container storage module, object storage module and policy library storage module, providing data encryption and decryption storage functions and data integrity protection functions. Among them, the data encryption and decryption storage function is mainly to encrypt and store account, container, object, and privacy policy data in each storage module. When each storage module needs to read its own data, a decryption algorithm is used to read the data, which can ensure the privacy of the data. The data integrity measurement function mainly performs initial measurement and real-time measurement of data by calling an algorithm, and stores the initial measurement result using the data encryption and decryption storage function to compare with the real-time measurement result to verify the integrity of the stored data;

The decryption algorithm adopts the SM2 algorithm, the SM2 algorithm is a more advanced and secure public key encryption algorithm, which is released by the State Password Administration, and the decryption steps are as follows:

(1) Convert the elliptic curve point A_1, check whether it is a point on the elliptic curve, if not, report an error and exit;
(2) Calculate the far point B_1, the calculation formula is as follows:

$$B_1 = [\varsigma]W_B \tag{4}$$

In formula (4), W_B refers to the point on the elliptic curve, ς refers to the random number generated by the random number generator.

If B_1 is not an infinity point, go to step (3), otherwise report an error and exit;

(3) Calculate the product of user 2's private key and the elliptic curve point D_1. The calculation formula is as follows:

$$R_2 D_1 = (\alpha_2,\ \beta_2) \tag{5}$$

In formula (5), R_2 refers to user 2's private key, $(\alpha_2,\ \beta_2)$ refers to user 2's SM3 password hash function coordinates.

(4) Calculate the plaintext threshold, if all are zero, report an error and exit;

(5) Calculate the length of the decrypted bit string, and calculate the threshold of the SM3 cipher hash function. When the threshold is not equal to the SM3 cipher hash function parameters, an error is reported to exit;
(6) Output plaintext.

The trusted communication module acts on each module that needs data interaction in the distributed execution method. It is combined with the trusted support module and the trusted storage module, for each data exchange node to verify each other whether the module execution code of each other has been maliciously tampered with and whether the running state is normal and reliable, so as to judge whether the communication node is trustworthy. When each module in the distributed execution method of the privacy policy performs data interaction, the steps are as follows:

1. The requester requests the receiver to request a metric log and a specific set of PCR values;
2. The receiver sends the metric log and a specific PCR value to the user, and also requests the metric log and a specific set of PCR values from the requester;
3. The requester invokes the algorithm to calculate the summary value of the metric log, and compares it with the received PCR value to verify whether the receiver's metric log is credible. After passing the verification, send its own measurement log, specific PCR value and service request to the receiver;
4. The receiver receives the metric log and the specific PCR value, and also verifies the metric log. If the verification is passed, it executes and responds to the service request. After the above interaction steps, each data interaction node can mutually verify whether each other's module execution code has been maliciously tampered with and whether the running state is normal and reliable.

The main idea of the method implementation scheme: the user submits the access request Request to the proxy server of the Swift cloud platform by using the remote access client program; the proxy server uses the distributed execution method of the privacy policy to judge the user's access request Request. And according to its judgment result, it responds to the user request. If the judgment result does not pass, it will reject the user request; if it passes, the proxy server will request Request according to the user's access request. Interact with object and container storage servers through the intranet Private to perform corresponding operations for data access. At the same time, in order to ensure the credible execution of the above-mentioned privacy policy, this method proposes a method for ensuring the implementation of the privacy policy, which is used to guarantee the credible implementation of the privacy policy in the above-mentioned distributed execution method.

3.3 Privacy Access Control

Design a data access control model based on blockchain. The entities included in the model are data owner, data requester, data storage center, attribute authority and blockchain network.

Data owners are users in the system who want to share data. Data owners need to formulate access policies for relevant shared files in advance. Only when the attribute

value of the data requester conforms to the access policy set by the user can decrypt the data shared by the user [9]. The user encrypts the data with the master key, encrypts the master key with CP-ABE, then stores the encrypted data and master key in the database, and publishes the corresponding information on the blockchain. Users need to update through the blockchain after updating the shared data.

The data requester tries to access the data stored in the database by the data owner in the system. The data requester requests the attribute authority to obtain the key of the corresponding attribute. If the attribute set conforms to the user's access policy, the data requester can successfully decrypt to obtain the master key, and then use the master key to decrypt to obtain the user information.

The data storage center provides remote data storage services for data owners, such as the common cloud storage services now, users can easily share data. In order to protect data security, data owners usually store data in the form of ciphertext. Unlike cloud storage, the data storage center in this paper does not need to control the access rights of data.

The attribute authority is a key part of the CP-ABE encryption algorithm. In this model, the attribute authority is also a key generation agency, responsible for generating and publishing the keys used by entities in the system. The attribute authority is in a distributed structure, each attribute authority is responsible for the generation of different attribute keys, and the attribute authority is independent of each other. The attribute authority includes the attribute key generation node and the attribute management node to implement access control.

In the system, the data owner is a trusted entity, which formulates access control policies and uploads data strictly in accordance with the protocol. The data requester is a semi-trusted entity in this system, and the requester may want to access user data that he does not have access rights while legally requesting user data. The requester may match the user's access control policy by means of joint attributes, so as to obtain the data that fails to meet the access policy. The data storage center is curious in this system, and user data has greater economic attractiveness to it, so users need to encrypt the stored data. Attribute authority plays a very important role in the system and is a trusted entity. In order to avoid security attacks due to excessive concentration of power, the system adopts distributed attribute authority. The model considers that there is no collusion between attribute authorities [10].

The model contains three smart contracts, namely property generation contract, key generation contract and data sharing contract.

The attribute generation contract is deployed by the attribute management nodes of each attribute authority. The attribute authorities are independent of each other, and the management attributes do not overlap. The main function of this contract is to record the attribute Token and valid time issued by the attribute management node to the data requester. The contract contains four functional functions, namely add data requester, remove data requester, add attribute and remove attribute.

Key generation contracts are deployed by key generation nodes authoritative for each property. The main function of this contract is to record the key generation status of the key generation node as the data requester. It contains four functional functions, namely, add data requester, remove data requester, add attribute key and remove attribute key.

Data sharing contracts are deployed by individual data owners. The data owner sends the ciphertext containing the master key and LocationM and its abstract to the data storage center through the data sharing contract and uploads it to the blockchain. The interaction between the data requester and the data storage center is recorded by the data sharing contract. It contains four functional functions, namely add digest, request digest, withdraw data visitor and policy update.

In cloud storage, it is very important to ensure the security of user data. Cloud storage often uses storage servers to store and manage users' shared data in plaintext, which easily compromises user data privacy. This solution uses the currently common AES encryption algorithm to encrypt user data, and encrypts the AES key with attribute-based encryption and stores it in the cloud.

The encryption process of the AES encryption algorithm is as follows:

(1) Byte replacement

Byte substitution is a non-linear permutation based on S-boxes, which are 16-byte matrices with 256 elements, each one byte in size. And all the elements are different, the elements are all hexadecimal representation.

(2) Line shift

Row shift is to change the position of elements in a cyclic left shift for each row of 4 bytes of the state matrix. The first row of the state matrix does not need to be operated, and the second to fourth rows require a left-shift byte to change the position of the element.

(3) Column mix

The column mixing transformation is realized by multiplying the state matrix $d(s)$ column by column with a fixed polynomial, as follows:

$$d'(s) = t(s)u(s) \bmod \left(s^4 + 1\right) \tag{6}$$

In formula (6), $t(s)$ represents the sequence set of hexadecimal bytes; $u(s)$ refers to the extended set of hexadecimal bytes; s refers to hexadecimal bytes.

The above formula can be expressed as a matrix:

$$\begin{bmatrix} d'(s)_0 \\ d'(s)_1 \\ d'(s)_2 \\ d'(s)_3 \end{bmatrix} = \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \begin{bmatrix} d(s)_0 \\ d(s)_1 \\ d(s)_2 \\ d(s)_3 \end{bmatrix} \tag{7}$$

In formula (7), $\begin{bmatrix} d'(s)_0 \\ d'(s)_1 \\ d'(s)_2 \\ d'(s)_3 \end{bmatrix}$ represents the four state values of $d'(s)$; $\begin{bmatrix} d(s)_0 \\ d(s)_1 \\ d(s)_2 \\ d(s)_3 \end{bmatrix}$ represents the four state values of $d(s)$.

(4) Round key encryption

The round key encryption operation is mainly to perform XOR operation on bytes.

The decryption process is still ten rounds, and the round function of each round is the inverse operation of the encryption operation: reverse byte replacement, reverse row shift, reverse column mixing and round key encryption.

Since the decryption of CP-ABE requires the decryptor to have an attribute key that complies with the access policy, it is guaranteed that an unauthorized party cannot restore the original message when it lacks at least one attribute key. The AES key is secured to ensure the safety of user-shared data stored in the cloud.At the same time, the information digest stored in the cloud by the user is published on the blockchain. The data requester downloads the relevant information from the cloud and calculates the digest value and compares it with the digest information recorded on the blockchain to verify whether the information has been replaced. At the same time, the scheme uses CP-ABE to realize that users can independently control the access rights of personal data. The user defines different access policies for different files to be shared. When the attributes of the data requester conform to the access policy set by the data owner, the ciphertext can be decrypted to obtain the content shared by the data owner. If the attribute of the data requester does not conform to the access policy defined by the data owner, the requester cannot decrypt the information through its own key, nor can it decrypt the information by combining the attribute key with other visitors. At the same time, the data owner can revoke the access rights of the designated data visitor through the smart contract, so this scheme can realize the fine-grained access control of the data.

4 Experimental Tests

4.1 Experimental Environment

In order to verify the effectiveness of the data privacy access control method based on the ciphertext policy attribute-based encryption algorithm, the experimental environment is set as: CPU: Intel Core i5-4200U 1.60 GHZ, memory: 4.0 GB, development tools: visual studio 2010 on Windows 7 for 64bit, keystore: MIRACL.

The system architecture of the alliance chain is adopted, so the experimental environment should include several organizations, and each organization should have endorsement nodes and submission nodes belonging to the organization. It should also include ordering nodes that provide ordering services. In the experimental environment, each node is maintained by a virtual machine, and each virtual machine is allocated 1 core processor and 1 GB of memory. In the experiment, the scenario of 3 organizations is simulated, and each organization has 3 submission nodes, including 1 endorsement node and 2 peer nodes. In addition, there are 4 ordering nodes in the experimental environment, and the ordering node and the submitting node run in the same virtual machine. In the attribute blockchain maintained by the endorsing nodes, the endorsing nodes take turns to play the role of the client in turn, and send requests to update the user access history information to the ordering nodes. The rotation interval of the endorsing nodes is 5 min.

4.2 Scalability, Fault Tolerance, No Third Party and Obtaining Authorization Test

The performance of the proposed method is tested in terms of scalability, fault tolerance, no third party and obtaining authorization, where " + " means better, "–" means weaker, and " + −" means case-by-case analysis. The test results are shown in Table 1.

Table 1. Test results of scalability, fault tolerance, no third party and obtaining authorization of the proposed method

Serial number	Project	The proposed method	The method of reference [1]	The method of reference [2]
1	Scalability	+	+	−
2	Fault tolerance rate	+	+ −	−
3	No third party	+	+	+
4	Get authorization	+	−	+

According to Table 1, it can be seen that the method of reference [1] and the method of reference [2] have weak links in the test process, while the scalability, fault tolerance, no third party and authorization of the proposed methods all show good performance. The proposed method has good scalability. As for the fault tolerance rate, since the blockchain is a distributed structure, information verification is carried out by all verification nodes in the same blockchain network, which can effectively resist the single-point failure problem and has a high fault tolerance rate. For the problem of no third party, blockchain-based distributed applications do not need to rely on third parties. Therefore, the proposed method is free of third parties. For obtaining authorization, the proposed method can also decrypt user data in real time on the premise that the data requester has obtained the attribute key in advance, so it can be considered as real-time authorization. In summary, the proposed method has the characteristics of good scalability, high fault tolerance, no third party, and quick authorization, it has good privacy access control effect.

4.3 Anti-attack Rate Test

In the experiment, replay attack, man in the middle attack, witch attack and conspiracy attack are applied. The test results of the attack resistance rate of the proposed method under the four attack modes are shown in Fig. 2.

According to the test results in Fig. 2, the proposed method has a resistance rate of 94.83% against replay attacks, 94.86% against man-in-the-middle attacks, 94.85% against sybil attacks, and 96.62% against collusion attacks. That is to say, the attack rate of the proposed method for the four attacks is higher than 94%, and it has good encryption performance.

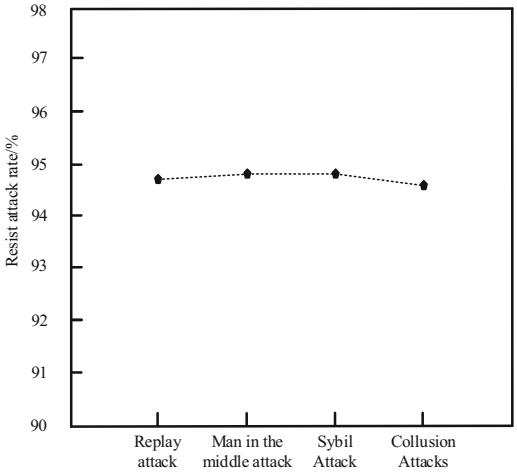

Fig. 2. Test results of attack resistance rate of the proposed method under four attack modes

4.4 Privacy Leakage Risk Rate Test

On this basis, the privacy access control performance of the proposed method is tested, mainly to test its privacy leakage risk. Taking the method of reference [1] and the method of reference [2] as a comparison method, the test results of the privacy leakage risk rate of different methods are obtained as shown in Fig. 3.

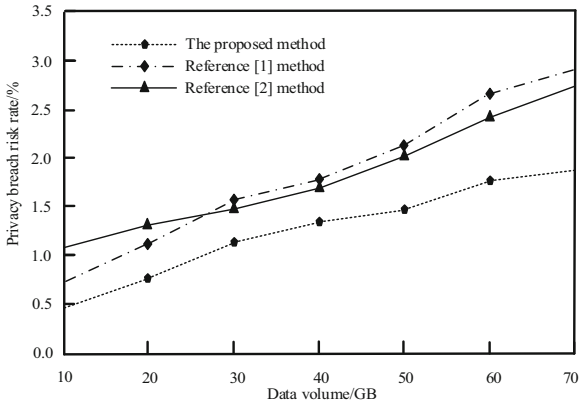

Fig. 3. Test results of privacy leakage risk rate for different methods

The test results in Fig. 3 show that while the amount of data continues to grow, the risk rates of privacy leakage of the three methods all increase to a certain extent. But overall, the privacy leakage risk rate of the proposed method is always lower than that of the method of reference [1] and the method of reference [2], indicating that the proposed method has a low risk of privacy leakage.The test results in Fig. 3 show that while the

amount of data continues to grow, the risk rates of privacy leakage of the three methods all increase to a certain extent. But overall, the privacy leakage risk rate of the proposed method is always lower than that of the method of reference [1] and the method of reference [2], indicating that the proposed method has a low risk of privacy leakage.

5 Conclusion

Data privacy protection in the cloud environment has become a bottleneck restricting the development of cloud services. From the perspective of cloud service scenarios, data privacy protection in cloud environment is studied, and a data privacy access control method based on ciphertext policy attribute-based encryption algorithm is designed. The cloud data privacy security issues are mainly attributed to the three stages of cloud service selection, access control and policy execution, and the privacy security of user data is improved and guaranteed from these three stages.

References

1. Feng, T., Pei, H., Ma, R., et al.: Blockchain data privacy access control based on searchable attribute encryption. Comput. Mater. Continua **66**(1), 871–890 (2020)
2. Yang, C., Tan, L., Shi, N., et al.: AuthPrivacyChain: a blockchain-based access control framework with privacy protection in cloud. IEEE Access **8**, 70604–70615 (2020)
3. Gao, P., Li, J., Liu, S.: An introduction to key technology in artificial intelligence and big data driven e-learning and e-education. Mob. Netw. Appl. **26**(5), 2123–2126 (2021)
4. Liu, S., Hu, R., Wu, J., et al.: Research on data classification and feature fusion method of cancer nuclei image based on deep learning. Int. J. Imaging Syst. Technol. **32**(3), 969–981 (2022)
5. Edemacu, K., Jang, B., Kim, J.W.: Efficient and expressive access control with revocation for privacy of PHR based on OBDD access structure. IEEE Access **8**, 18546–18557 (2020)
6. Mi, B., Long, P., Liu, Y., et al.: Balancing access control and privacy for data deduplication via functional encryption. Math. Probl. Eng.Probl. Eng. **2020**(4), 1–11 (2020)
7. Mehta, V., Gooch, D., Bandara, A., et al.: Privacy care: a tangible interaction framework for privacy management. ACM Trans. Internet Technol. **21**(1), 1–32 (2021)
8. Alemany, J., Val, E.D., García-Fornes, A.: A review of privacy decision-making mechanisms in online social networks. ACM Comput. Surv. (CSUR) **55**(2), 1–32 (2022)
9. Iwaya, L.H., Ahmad, A., Babar, M.A.: Security and privacy for mHealth and uHealth systems: a systematic mapping study. IEEE Access **8**, 150081–150112 (2020)
10. Ma, X.: User information privacy query access control based on BP neural network. Comput. Simul. **37**(7), 341–345 (2020)