




Research Article

Cloud Storage Data Access Control Scheme Based on Blockchain and Attribute-Based Encryption

Xiaodong Yang ¹, Aijia Chen ¹, Zhisong Wang,¹ and Shudong Li ²

¹College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China

²Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China

Correspondence should be addressed to Xiaodong Yang; y200888@163.com and Shudong Li; lishudong@gzhu.edu.cn

Received 19 November 2021; Revised 23 December 2021; Accepted 30 March 2022; Published 11 May 2022

Academic Editor: Yuling Chen

Copyright © 2022 Xiaodong Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cloud storage is a popular model of the application in various fields, and the security of storage data and access permission have been widely considered. Attribute-based encryption (ABE) provides fine-grained user access control and ensures data confidentiality. However, current ABE access control schemes rely on trusted cloud servers and provide a low level of security. To solve these problems of traditional encryption schemes, we propose a blockchain-based and ABE cloud storage data access control scheme. In this article, blockchain and smart contract technology are the core elements to ensure data integrity and build a decentralized verification method for outsourcing results. This application can minimize the reliance on servers in the cloud environment. Based on the ciphertext-policy ABE algorithm, the proposed scheme supports a hidden access policy to avoid the risk of privacy leakage. In addition, we adopt outsourcing technology and predetected decryption algorithms to reduce the computational overhead of local and outsourced servers. Security analysis and performance evaluation show that our proposed scheme has high computational efficiency and satisfies the condition of indistinguishability under the chosen-ciphertext attacks.

1. Introduction

Cloud storage technology uses the storage space of cloud servers to provide powerful data storage capability [1]. Data owners can overcome the obstacle of restricted storage resources at user terminals by storing data in the cloud. Therefore, cloud storage has become more popular in various specific industries in recent years, such as the Internet of Things (IoT) [2, 3], the Industrial Internet of Things environment [4], and electronic health records [5, 6]. However, the data collected by cloud servers and IoT devices face many attacks [7] during data transmission and storage. Meanwhile, sensitive data are vulnerable to tampering or forgery attacks during the transmission via public channels, which exposes users' private information to the risk of being leaked. Therefore, it is critical to consider privacy protection and data confidentiality in the network. In the most typical schemes, encryption technology is adopted to achieve data confidentiality and privacy. To provide more detailed

privacy protection, some researchers introduce the most recent privacy protection technologies in their schemes. For instance, a location privacy protection scheme [8] anonymizes the source location, which contains significant information about the target being observed and tracked. Moreover, a homomorphic encryption scheme with higher performance [9] is proposed to achieve privacy protection of data stored in the central server.

Although the encryption mechanism can guarantee the confidentiality and privacy of the data, it does not ensure that the data are legally obtained. In cloud storage applications, the data stored in the cloud server cannot be fully controlled by the data owner. To prevent malicious users and cloud server providers from accessing data, a trusted access control mechanism is also essential.

The CP-ABE [10] not only provides data confidentiality but also allows fine-grained and flexible access control to improve the security of the data. However, the traditional CP-ABE scheme [10, 11] has some drawbacks in practical

applications. For example, the access control policy in the CP-ABE is constructed by attribute information-related users, which may contain private information about the user's identity. Second, attribute-based encryption algorithms frequently use a large number of bilinear pair computations, significantly increasing the encryption and decryption computational overhead. To reduce computational costs, on the one hand, an increasing number of schemes outsource decryption operations to third-party servers. However, few of these systems consider the correctness of calculation results from cloud servers. On the other hand, most access control schemes on cloud platforms are established using prime-order bilinearity to reduce the computational burden. This design's reduced computational burden comes at the expense of lower security, so it can only satisfy indistinguishability under chosen-plaintext attack (IND-CPA). Although there are already some schemes that can partially solve the above problems, we still need to consider some detailed and in-depth issues. The existing cloud storage access control scheme is designed based on the traditional cloud server, which increases the trusted dependence on the cloud server. Unfortunately, semitrusted cloud servers are curious about the processed data while executing user commands. If the cloud server fails unpredictably or is maliciously attacked and outputs incorrect results, it may cause users to obtain incorrect data.

Blockchain technology [12] is a widely emerging technology based on distributed ledgers that has the advantages of decentralization. However, at the same time, due to the openness of blockchain, data security and supervision are also faced with challenges [13, 14]. Therefore, the combination of blockchain technology and traditional access control is a promising structure. Blockchain technology can enhance the reliability of traditional schemes, and the encryption mechanism of the scheme can protect the data security of the blockchain. In this article, we are committed to establishing a reliable access control mechanism in an untrusted cloud environment. We propose a cloud storage access control scheme based on blockchain and attribute-based encryption, which realizes data verification and ensures the verifiability of the outsourced decryption results and the integrity of the cloud storage data in a decentralized way.

The main contributions of our proposed program are as follows:

- (i) The support of hidden access control policies reduces the risk of user privacy information disclosure in traditional CP-ABE.
- (ii) The use of smart contracts deployed on the consortium blockchain can achieve a decentralized verifiable outsourcing scheme while ensuring the integrity of data in the cloud.
- (iii) The dependence on fully trusted cloud servers in traditional cloud server-based schemes is removed by introducing blockchain technology.
- (iv) Our scheme is proven to meet CCA security under the random oracle model, which has stronger

security than similar schemes. Performance analysis shows that the new scheme has comparable computational overhead.

The rest of the article is organized as follows. Section 2 introduces the related work. Preliminary knowledge related to our scheme is described in Section 3. In Section 4, we present the system model, security model, scheme framework, and detailed construction of the proposed scheme. The correctness analysis is given in Section 5. In Section 6, we provide security analysis and security proof of the new scheme. In Section 7, we discuss the performance analysis and computational efficiency of our scheme. The work of this scheme is concluded, and the outlook is presented in Section 8.

2. Related Work

To overcome the problem of multiperson sharing of encrypted data, an attribute-based encryption system (ABE) [15] was proposed as a one-to-many encryption mechanism. More specifically, ciphertext-policy attribute-based encryption (CP-ABE) [10] allows the data owner to refine the user authority of the data visitor to the attribute level by setting a policy. In other words, CP-ABE can achieve effective fine-grained access control under the condition of ensuring data security.

However, the traditional CP-ABE Schemes [16, 17] usually publish the access policy in the form of plaintext. Anyone who obtains the ciphertext (including cloud servers) can infer part of the secret information included in the ciphertext, endangering the user's identity privacy. In addition, sensitive data must also be protected as private data in specific fields.

To address the above issues, Kapadia et al. [18] proposed a policy-hiding CP-ABE scheme. However, an online semitrusted server was introduced in [18] to reencrypt the ciphertext for each user, thus making the server a bottleneck in the entire system. Nishide et al. [19] developed two CP-ABE schemes to hide the policy, which express the access control policy through AND logic with wildcards. Based on the decisional assumption of subgroups, Lai et al. [20] suggested an adaptively secure policy hiding the CP-ABE technique over a bilinear group of combinatorial orders. Although the scheme in [20] improves security, the computational cost grows with the increase of the attributes. Hur [21] constructed a scheme that supports arbitrary expressions with monotonicity and blinds the access policy within the ciphertext. However, this scheme is proven to be secure using the generic group model, which is normally considered heuristically rather than provably secure. Afterwards, Helil Rahman [22] constructed a CP-ABE access control scheme based on the scheme in [21]. We introduce an additional entity (the SDS monitor) in [22] to handle the problem of sensitive dataset constraints, but the policy is disclosed for all entities. Song et al. [23] made improvements to the access tree on the basis of the scheme in [24] to realize policy hiding based on the access tree. Through the application of secret sharing in "and," "or" and "threshold,"

attribute values with permission are hidden in all attribute values of the system. However, as the expression ability of the access structure grows, the communication overhead also increases.

To reduce the overhead of a large number of bilinear pairings required for the CP-ABE decryption calculation, Green et al. [25] proposed a scheme with outsourced decryption. In their article, the outsourcing server uses a transformation key for decryption, which is generated by the data user. However, their scheme lacks a verification mechanism for the calculation results of the outsourcing server. Then, on the basis of the scheme in [25], Lai et al. [26] verified the result returned by the outsourcing server by adding a ciphertext component. However, at the same time, this method doubles the ciphertext length of the ABE-type and El Gamal-type encryption systems. In recent years, with the development of fog computing, fog nodes have been widely used in cloud environments. Li et al. [27] presented a verifiable outsourced multiauthorization access control method that delegated most encryption and decryption work to fog nodes. This scheme can lighten the user's processing load and verify the reliability of outsourced computing outputs. In fog-enhanced IoT systems, an access control scheme with hidden access structures and outsourcing computation was presented by [28], which uses fog nodes to conduct outsourcing decryption and verification procedures. Lin et al. [29] invented a new attribute-based scheme combined with symmetric encryption technology to achieve efficient verifiability. In addition, they presented a verifiable unified model for the OD-ABE. However, all of the abovementioned verifiable outsourcing schemes meet the CPA security requirements. A verifiable hidden policy CP-ABE with a decryption testing scheme (VHPDT) was proposed by Zhao et al. [30], which is CCA-secure. Meanwhile, the VHPDT scheme introduces a predetection algorithm to increase the efficiency of the decryption. However, this scheme does not consider the integrity verification of the data and needs to rely on trusted cloud servers. However, cloud servers cannot be completely trusted, and dangers such as user data leakage and tampering will persist.

Blockchain technology [12] is an emerging technology based on distributed ledgers that has the advantages of decentralization. Many systems [31–34] introduce blockchain into the traditional cloud server-based structure to better realize decentralized security schemes. Rahulamathavan et al. [32] proposed combining blockchain technology with ABE to realize data confidentiality and privacy protection. However, the large amount of computing overhead generated by ABE is not suitable for the resource-constrained IoT environment. Zhang et al. [33] introduced blockchain-based smart contract technology and designed a BaDS scheme in the IoT, which not only reduces the cost of decryption but also improves the flexibility of traditional CP-ABE for access control. A blockchain-based outsourcing verifiable CP-ABE scheme was offered by Zhang [34], which uses smart contracts to achieve verifiability of the outsourcing results. However, decrypting and obtaining plaintext by smart contracts will reduce the security of the system.

3. Preliminary Knowledge

3.1. Composite-Order Bilinear Group. Assuming that φ is a group generation algorithm, the input λ is a security parameter, and the output $(N = p_1 p_2 p_3, G, G_T, \hat{e})$ is a tuple, where N is the product of three prime numbers p_1, p_2 , and p_3 ; G and G_T are cyclic groups with order N ; $\hat{e}: G \times G \rightarrow G_T$ is a bilinear map satisfying the following conditions:

- (1) **Bilinearity:** for any $g_0, g_1 \in G$, $c, d \in \mathbb{Z}_N$, we have $e(g_0^c, g_1^d) = e(g_0, g_1)^{cd}$.
- (2) **Nondegeneracy:** if $x \in G$, then $e(x, x)$ has the order N in G_T .
- (3) **Computability:** if $\hat{e}: G \times G \rightarrow G_T$, then operations in G and G_T are effectively computable in polynomial time, and G and G_T are bilinear groups.
- (4) **Orthogonality:** G_{p_1}, G_{p_2} , and G_{p_3} are three subgroups of G_0 , with the order of p_1, p_2 , and p_3 , respectively. The orthogonality of the subgroups can be known as follows:
 - (a) For any $h_{p_1} \in G_{p_1}$ and $h_{p_2} \in G_{p_2}$, then $e(h_{p_1}, h_{p_2}) = 1$.
 - (b) For any $h_{p_1} \in G_{p_1}$ and $h_{p_2} \in G_{p_2}$, where $a, b, c, d \in \mathbb{Z}_N$, equation $e(h_{p_1}^a h_{p_2}^b, h_{p_1}^c h_{p_2}^d) = e(h_{p_1}, h_{p_2})^{ab} = e(h_{p_1}, h_{p_2})^{cd}$ holds.

3.2. Discrete Logarithm (DL) Problem. Let G_0 be a multiplicative cyclic group of order p_1 and g_1 be the generator of G_0 . Given a tuple $(g_1, \Delta = g_1^x)$, where $\Delta \in G_0$, the DL problem has difficulty calculating $x \in \mathbb{Z}_N$.

3.3. Blockchain and Smart Contracts. The essential function of blockchain technology is a distributed ledger that cannot be tampered with and counterfeited [12]. Blockchain technology joins data blocks in chronological order to form a chain data structure and uses cryptography to assure the chain's immutability and security. Moreover, blockchain encourages network nodes to participate in and jointly maintain chain data by setting up incentive mechanisms to provide a reward. The consensus mechanism is adopted to ensure the fairness of transactions, which is based on multiparty consensus and will not be undermined by the complicity of a few malicious nodes. Therefore, blockchain can be used as a low-cost and highly reliable infrastructure. Blockchain is deployed in the forms of public blockchain, private blockchain, and consortium blockchain. The public blockchain is a mode in which any node is open to anyone. This mode allows everyone to participate in the calculation of this block, and anyone can download and obtain the full blockchain data. The private blockchain is a private chain in which only licensed nodes can be involved and view all data. Consortium blockchain means that the permissions of each node participating are completely equal. Without total mutual trust, each node can realize the trustworthy exchange of data, but each node often has an associated entity organization that may only join or leave the network after

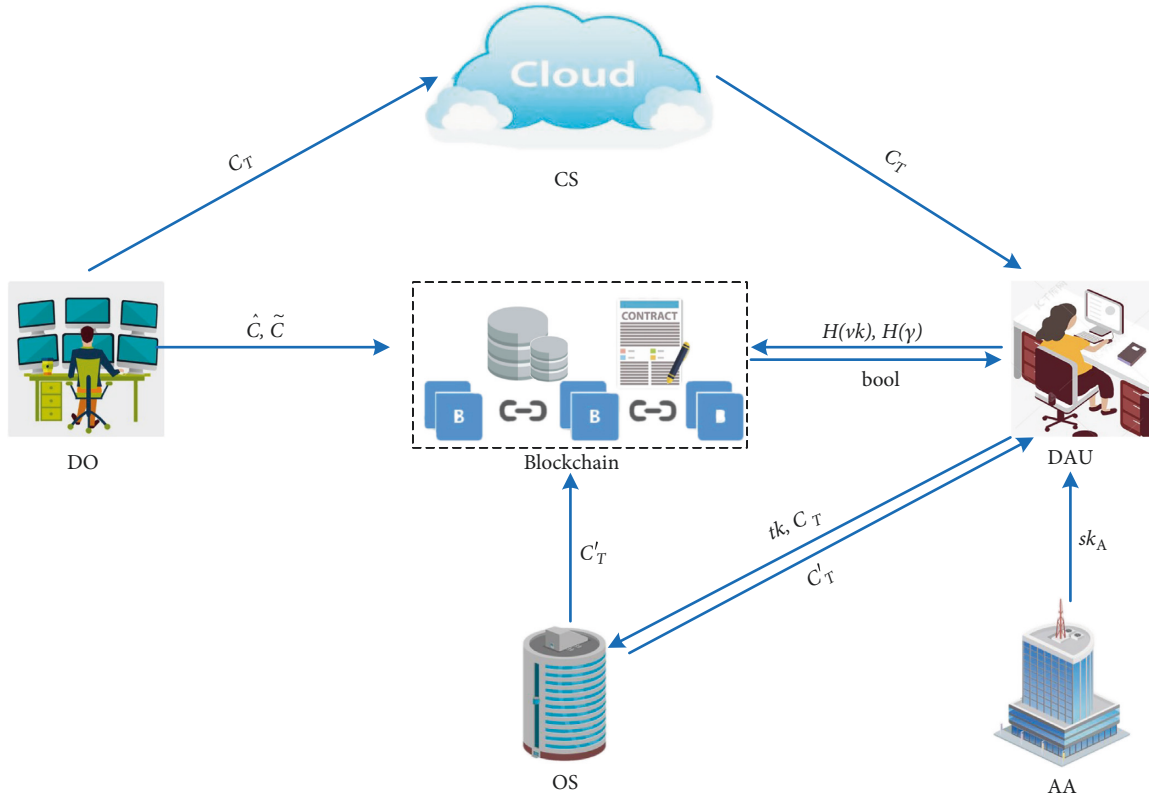


FIGURE 1: System model.

being authorized. Compared with the public blockchain, the consortium blockchain maintains the characteristics of decentralization and enhances the control of the participating members.

A smart contract is an automatic piece of code deployed on the blockchain with a unique address [35]. The initializer can establish a smart contract and save it as a transaction on the blockchain platform. When a transaction in the contract is triggered, the contract will automatically execute predefined content according to the script, such as executing relevant calculations. Finally, the output and status information of the transaction are recorded in the blockchain as transactions. In our structure, we employ smart contracts to create interfaces for the blockchain application layer and verify operations through the interaction of cloud servers with smart contracts instead of using semitrusted servers.

4. Our Cloud Storage Data Access Control Scheme Based on Blockchain and Attribute-Based Encryption

4.1. System Model. Figure 1 depicts the framework of our data access control system, which includes six entities: Attribute Authority, Data Owner, Cloud Server, Data Accessing Users, Blockchain, and Outsourcing Server. The functions of various entities are described as follows:

- (i) The *Attribute Authority (AA)* is responsible for setting up the system and generating the users' private keys.

- (ii) The *Data Owner (DO)* calculates the hash of the initial data and parameters used for authentication and uploads these components to the blockchain platform. Then, the DO generates the ciphertext by encrypting the plaintext according to the access policy and sends it to the cloud server for storage.
- (iii) The *Cloud Server (CS)* is a semitrusted entity that stores data ciphertext.
- (iv) The *Data Accessing User (DAU)* is initially involved in generating a key that is used by the outsourcing server for decryption. After receiving the storage address returned by the cloud server, the DAU is responsible for computing parameters and decrypting. After obtaining the plaintext, the DAU verifies the integrity of the data through the computation.
- (v) *Blockchain.* We use a consortium blockchain with smart contracts deployed. The blockchain platform is responsible for storing verification components and smart contracts, ensuring the correctness of the outsourcing decryption result.
- (vi) The *Outsource Server (OS)* is responsible for detecting the attributes of the accessing user and obtaining the semiciphertext through decryption.

4.2. Security Model. To fulfil the confidentiality and verifiability of the proposed scheme, we define the security model of our scheme by the following two security games.

Game 1 (confidentiality): for our scheme, we define an indistinguishable game under the chosen-ciphertext attack (IND-CCA) that includes an adversary Algorithm A and a challenge Algorithm B .

Initialization phase: B runs Setup(1^λ) to produce the system public key pk and the system master private key msk . Then, B sends pk to A and retains msk .

Inquiry phase 1: A adaptively asks B for the private key of the attribute set Λ , and the private key can be requested repeatedly. B runs KeyGen(pk, msk, Λ) and returns sk_Λ to A .

Challenge phase: A sends equal-length messages M_0 and M_1 as well as access structures W_0 and W_1 to B . B selects $\xi \in \{0, 1\}$ and runs Encrypt(pk, m, W) to generate challenge ciphertext C^* . Finally, B sends C^* to A .

Inquiry phase 2: this is similar to inquiry phase 1, but A cannot ask for the messages M_0 and M_1 .

Guess: A outputs the guess $\xi' \in \{0, 1\}$ of the challenge ciphertext C^* . If $\xi = \xi'$, then B outputs 1, which means that A wins Game 1 with a probability of $\text{Adv} = |\Pr[\xi = \xi'] - 1/2|$.

Theorem 1. *If there is no polynomial-time adversary to attack the above security model with a nonnegligible probability advantage, then our proposed scheme is IND-CCA.*

Game 2 (verifiable): We use the interactive game between adversary F and challenger C to prove the verifiability of our scheme supporting the hidden strategy. The process is as follows:

Initialization phase: C runs the Setup algorithm to produce the master key msk and the system public key pk , while pk is sent to F .

Challenge phase: F asks for the decryption key by specifying an arbitrary set of attributes Λ to be sent to C for inquiry. Then, C performs a key generation algorithm based on the attribute set Λ to generate a decryption key sk . Finally, sk is returned to adversary F .

Output phase: F outputs an access structure W that satisfies the attribute set Λ and a tuple $(C'_T, tk, C_{k1}, C_{k2}, \Delta)$. C executes the preauthentication algorithm to obtain the session key nk_1, nk_2 . If $nk_1 \neq nk_2$, then we claim that F wins the game. We define $\text{Pr}[F \text{ wins}]$ to denote the advantage of F winning the game.

Theorem 2. *If there is a polynomial adversary F who can win the above interactive game with the advantage $\text{Pr}[F \text{ wins}]$, then our attribute-based encryption scheme with the hidden strategy can be considered to be verifiable.*

4.3. Scheme Framework. The operational flow of the cloud storage data access control scheme based on blockchain and

attribute-based encryption is shown in Figure 2, and the specific implementation details of this scheme are as follows.

4.4. Scheme Construction

4.4.1. System Setup. The credible attribute authorization centre (AA) executes the system setup algorithm. is a group generation algorithm that outputs tuple $(N = p_1 p_2 p_3, G, G_T, \hat{e})$. AA first selects a security parameter λ and runs the algorithm $\varphi(\lambda)$ to obtain the system parameters $(N = p_1 p_2 p_3, G_0, G_T, \hat{e})$, where G_0 and G_T are two cyclic groups of order N , and p_1, p_2 , and p_3 are three different prime numbers. G_{p_1}, G_{p_2} , and G_{p_3} are three subgroups from G_0 , whose generators are g_1, g_2 , and g_3 , respectively. We suppose that $U = \{\text{att}_1, \text{att}_2, \dots, \text{att}_n\}$ is a system attribute set and $S_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,j}\}$ is the value set of the attribute att_i . For any attribute att_i in the system, AA generates a public key pk and a master key msk according to the following steps:

- (1) AA chooses two hash functions in cryptography $H: \{0, 1\}^* \rightarrow Z_N$ and $H_0: G_0 \rightarrow Z_N^*$, which are anticollision.
- (2) For any attribute att_i in the system, AA randomly selects $x_{i,j} \in Z_N^*$ and $Q_{i,j} \in G_{p_3}$ and calculates $A_{i,j} = g_1^{1/x_{i,j}} Q_{i,j}$, where $i \in (1, 2, \dots, n), j \in (1, 2, \dots, n_i)$.
- (3) AA randomly selects $\beta_0, \beta \in Z_N^*$ and $Q_0 \in G_{p_3}$ and then calculates $Y_0 = e(g_1, g_1)^{\beta_0}$ and $Y = e(g_1, g_1)^\beta$.
- (4) AA defines a key distribution function KF that maps the session key to a stream of bits of length κ and two parameters ω and ν that belong to G_{p_3} .
- (5) AA publishes the public key $pk = (A_0, g_3, \{A_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq n_i}, Y_0, Y, KF, \omega, \nu, \kappa, H, H_0)$ and keeps the master private key $msk = (g_1, \{x_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq n_i}, \beta_0, \beta)$ secretly.

4.4.2. Key Generation. According to the attribute list Λ of DAU, AA randomly selects $\lambda_i \in Z_N^*$ for any attribute $i (1 \leq i \leq k)$ and calculates $K_0 = g_1^{\beta_0 - \sum_{i=1}^k \lambda_i}$, $K = g_1^{\beta - \sum_{i=1}^k \lambda_i}$ and $K_i = g_1^{\lambda_i x_{i,j}}$. Then, AA sends the generated private key $sk_\Lambda = (K_0, K, \{K_i\}_{1 \leq i \leq k})$ to DAU.

4.4.3. Verification Component Generation. The data owner (DO) performs the following operations to generate and upload verification components.

- (1) The DO randomly selects $s \in Z_N$ and a session key $nk = Y_0^s = e(g_1, g_1)^{\beta_0 s}$ and uses the key distribution function $KF(nk, \kappa) = vk \parallel \gamma$ defined by AA, where γ is a random value and vk is the verification key. Then, the DO calculates $\hat{C} = \omega^{H(vk)} \gamma^{H(\gamma)}$, which is used to verify the outsourcing decryption result.

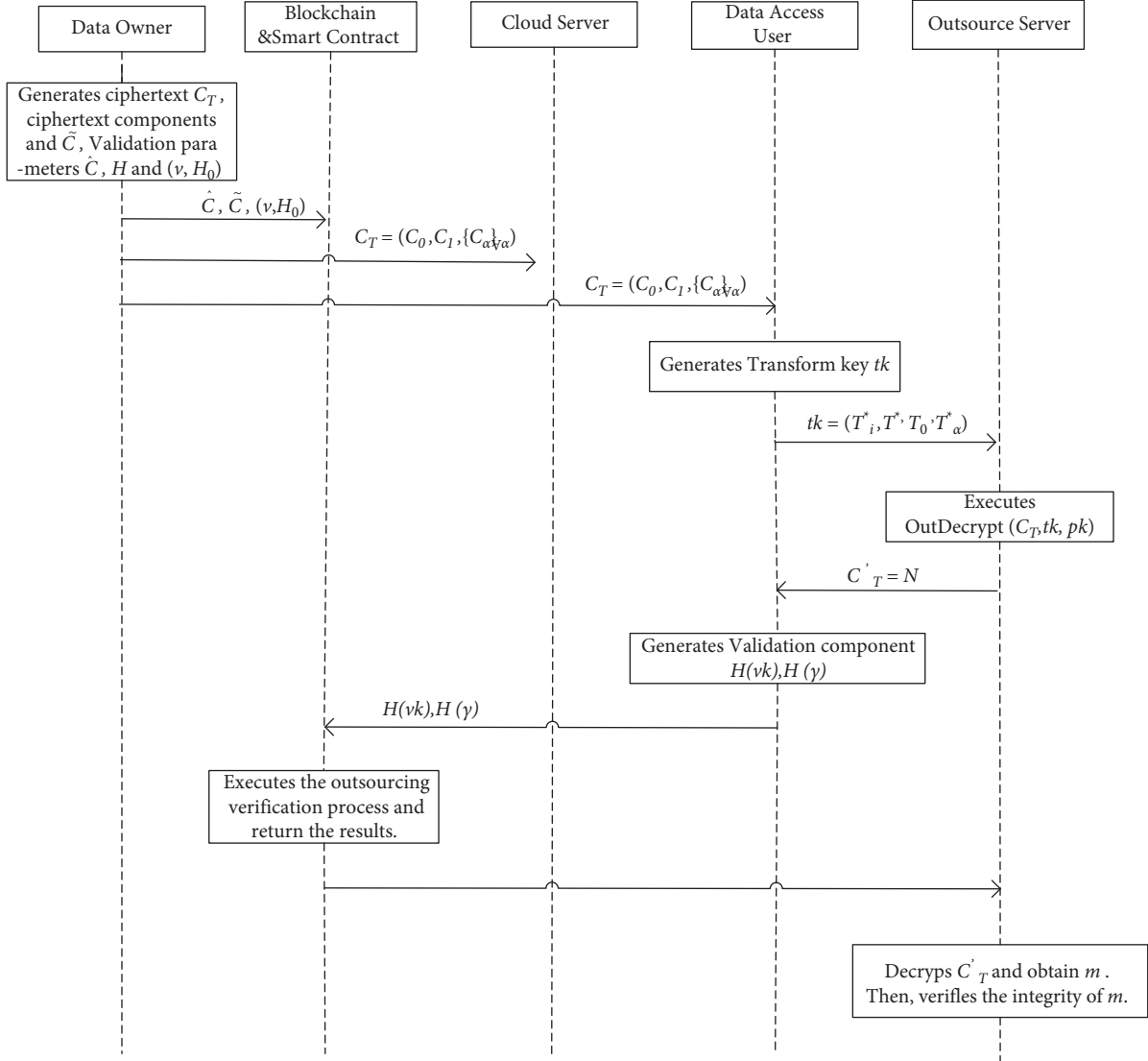


FIGURE 2: Framework of the proposed system.

- (2) The DO computes $\tilde{C} = v^{H_0(m)}$ and uploads to the blockchain platform. The stored addresses Add_m and (v, H_0) are sent to the smart contract as verification components.

4.4.4. Data Encryption. We adopt the access structure used in Zhao et al. scheme [30]. The DO performs the following operations with the access policy W to encrypt plaintext M .

- (1) The DO selects a random element $Q'_0 \in G_{p_3}$ and then calculates $C_0 = A_0^s Q'_0$ and $C_1 = mY^s$.
- (2) The DO sets the secret value s as the root node's value of the access tree. Then, the status of leaf nodes is set to read. Apart from leaf nodes, the status of all child nodes is set to unread. Later, the DO performs a recursive operation for each node with an unread state:

- (a) If the nonleaf node represents a logic "AND," then DO sets s_i for the $u - 1$ previous nodes of its children. Then, the value of the last leaf node is calculated by $s_u = s - \sum_{i=1}^{u-1} s_i$.
- (b) If the nonleaf node delegates a logic "OR," then DO sets s as the value of all child nodes, while the state of these nodes is set to read.
- (c) If the nonleaf node expresses the "threshold" with a threshold value h , then the DO randomly generates a polynomial f of degree $h - 1$. Meanwhile, the polynomial satisfies $f(0) = s$ and assigns the value of $f(i)$ to the i th child node.

- (3) The DO enforces operations to hide the policy. For simplicity, the parent node of any leaf node is named PNode. Suppose a PNode α exists, which is assigned the secret value s_α . Γ_α represents a subtree in which α is the root node, and all leaf nodes are indicated by a set S_{Γ_α} . For each attribute att_i , DO calculates $C_{i,j}$

according to different conditions. When an attribute $att_i \in \Gamma_\alpha$ and the value $v_{i,j} \notin S_{\Gamma_\alpha}$, the DO randomly selects $s_{i,j} \in Z_N^*$ and $Q'_{i,j} \in G_{p_3}$ and calculates $C_{i,j} = A_{i,j}^{s_{i,j}} Q'_{i,j}$. Otherwise, DO calculates $C_{i,j} = A_{i,j}^{s_{i,j}} Q'_{i,j}$.

- (4) The DO randomly selects $Q_\alpha \in G_{p_3}$, calculates $\bar{C}_\alpha = A_0^{s_\alpha} Q_\alpha$ and $I_\alpha = Y^{s_\alpha}$ for each PNode α , and obtains the component of the ciphertext $C_\alpha = (\bar{C}_\alpha, I_\alpha, \{C_{i,j}\}_{(1 \leq i \leq n, 1 \leq j \leq n)})$.
- (5) The DO obtains the entire ciphertext $C_T = (C_0, C_1, \{C_\alpha\}_{v \in \alpha})$ and sends it to the CS for storage.

4.4.5. Transformation Key Generation. DAU randomly chooses a factor $y \in Z_N^*$ and calculates $T_i^* = T_i^{1/y}$, $T^* = T^{1/y}$, $T_0^* = T_0^{1/y}$, and $I_\alpha^* = I_\alpha^{1/y}$. Later, DAU sends the transformation key $tk = (T_i^*, T^*, T_0^*, I_\alpha^*)$ and semidecrypted ciphertext C_T' to the outsourcing server OS.

$$\text{PreDecNode}(\beta) = \begin{cases} \prod_{i=1}^u \text{PreDecNode}(\text{child}(\beta, i)), & \text{structure}(\beta) = \text{AND}, \\ \text{PreDecNode}(\text{child}(\beta, i)), & \text{structure}(\beta) = \text{OR}, \\ \prod_{i=1}^h \text{PreDecNode}(\text{child}(\beta, i))^{\Delta_{i,\beta_0}}, & \text{structure}(\beta) = \text{Threshold}. \end{cases} \quad (1)$$

Finally, OS calculates $\omega_\chi = I_\alpha^* / e(C_\alpha, T^*) \text{PreDecNode}(\alpha)$.

- (2) Only when $\omega_\chi = 1$, does the OS further calculate $N = e(C_0, T_0^*) \text{PreDecNode}(\text{root}(W))$ in the decryption phrase. Then, the OS sends the semidecrypted ciphertext $C_T' = N$ to the DAU.

The preauthentication DAU obtains the semidecrypted ciphertext C_T' and generates the computed values $H(vk)$ and $H(\gamma)$ to complete the preauthentication work.

- (1) DAU uses the blinding factor γ and computes the session key $nk = N^{-\gamma} = e(g_1, g_1)^{s\beta_0}$.
- (2) DAU executes $KF(nk, \kappa) = vk \parallel \gamma$ mapping the session key to a stream of bits of length κ . Finally, the DAU calculates $H(vk)$ and $H(\gamma)$ sends it to the smart contract.

4.4.7. Outsourcing Verification. Receiving the elements $H(vk)$ and $H(\gamma)$ from the DAU, the smart contract computes $\omega^{H(vk)} \gamma^{H(\gamma)}$. If equation $\omega^{H(vk)} \gamma^{H(\gamma)} = \hat{C}$ holds, then the smart contract outputs $\text{bool} = 1$. Otherwise, the algorithm is terminated.

4.4.8. Decryption and Integrity Verification. If DAU receives $\text{bool} = 1$, then the semidecrypted ciphertext C_T' computed by the OS is not fake. Then, the steps of decryption and verification by the DAU are as follows:

- (1) DAU utilizes γ to compute plaintext $m = \hat{C} \cdot N^\gamma$.

4.4.6. Outsourcing Decryption. Execution by the outsourcing server OS. The algorithm is divided into an attribute detection phase and a decryption phase. The attribute detection phase is to preeliminate the attribute values in the private key that are unable to meet the access policy. This design can avoid bottom-up recursive decryption to reduce computational overhead. Only after passing the attribute checking can the algorithm proceed to the decryption phase.

- (1) The OS runs different functions according to different nodes in the access structure to detect the value. If a node is PNode α , the OS runs $\text{PreDecNode}(\alpha) = \prod_{i=1}^k e(C_{i,j}, T_i^*)$. Likewise, if a node is a normal node β , according to the structure of "OR", "AND" and "Threshold" in the access structure, then the OS runs $\text{PreDecNode}(\beta)$.

- (2) DAU computes $\tilde{C}' = v^{H_0(m)}$ and determines whether the computed \tilde{C}' equals \tilde{C} . If equation $\tilde{C}' = \tilde{C}$ holds, then the ciphertext stored on the cloud is proved completely.

5. Correctness Analysis

5.1. Correctness of Data Decryption. Here, we verify the correctness of the outsourcing decryption algorithm (executed by OS) and decryption algorithm (by DAU).

Receiving $tk = (T_i^*, T^*, T_0^*, I_\alpha^*)$ sent from the user, the OS executes attribute detection. The OS judges whether the user access structure satisfies all s_α values through the ω_χ result value calculated in the attribute detection phrase. The calculation equation is as follows:

$$\begin{aligned} \omega_\chi &= \frac{I_\alpha^*}{e(C_\alpha, T^*) \text{PreDecNode}(\alpha)} \\ &= \frac{\gamma^{s_\alpha/y}}{e(A_0^{s_\alpha} Q_\alpha, g_1^{\beta-\lambda/y}) e(g_1, g_1)^{\lambda s_\alpha/y}}, \\ &= \frac{e(g_1, g_1)^{\beta s_\alpha/y}}{e(A_0^{s_\alpha} Q_\alpha, g_1^{\beta-\lambda/y}) e(g_1, g_1)^{\lambda s_\alpha/y}} \\ &= \frac{e(g_1, g_1)^{\beta s_\alpha/y}}{e(g_1, g_1)^{s_\alpha(\beta-\lambda)/y} e(g_1, g_1)^{\lambda s_\alpha/y}} = 1. \end{aligned} \quad (2)$$

Only when the user's attributes pass the detection, can the OS obtain $\omega_\chi = 1$; otherwise, ω_χ is a random value. After

receiving $\omega_\chi = 1$, the OS uses tk to calculate C'_T , and the calculation equation is as follows:

$$\begin{aligned}
N &= e(C_0, T_0^*) \text{Pre Dec Node}(\text{root}(W)) \\
&= e(A_0^s Q_0', T_0^{1/y}) \text{Pre Dec Node}(\text{root}(W)) \\
&= e(g_0^s Q_0^s Q_0', g_1^{\beta_0 - \lambda/y}) e(g_1, g_1)^{\lambda s/y} \\
&= e(g_1, g_1)^{s(\beta_0 - \lambda)/y} e(g_1, g_1)^{\lambda s/y} \\
&= e(g_1, g_1)^{s\beta_0/y}.
\end{aligned} \tag{3}$$

DAU receives the C'_T sent from the OS and then calculates $nk = N^{-y} = e(g_1, g_1)^{s\beta_0}$ and $KF(nk, \kappa) = vk\|\gamma$. The smart contract verifies whether semidecrypted ciphertext C'_T is valid. If equation $\omega^{H(vk)\gamma^{H(\gamma)}} = \tilde{C}$ holds, then the decryption result from the OS is correct. Then, DAU using N , \tilde{C} and y recover the plaintext by the following:

$$\begin{aligned}
\tilde{C} \cdot N^y &= \frac{mY^s}{e(g_1, g_1)^{s\beta_0}} \\
&= \frac{me(g_1, g_1)^{s\beta_0}}{e(g_1, g_1)^{s\beta_0}} = m.
\end{aligned} \tag{4}$$

5.2. Integrity of Cloud Data. After the DAU obtains the plaintext, he or she calculates $\tilde{C}' = v^{H_0(m)}$ and verifies that \tilde{C}' is equal to the \tilde{C} stored on the blockchain. If $\tilde{C}' \neq \tilde{C}$, then the tampering of the ciphertext by the cloud server is demonstrated.

6. Security Analysis

6.1. Confidentiality. Data confidentiality of our scheme relies on the security of the attribute encryption system. This section proves Theorem 1 based on the security model in Section 4.2.

Theorem 3. *If there is no polynomial-time adversary that can attack the scheme of [30] with a nonnegligible advantage, then no polynomial adversary A can break the scheme of this article with a nonnegligible advantage.*

Proof. Based on the proof method in Scheme [30], we prove that the confidentiality of our scheme satisfies security under a chosen-ciphertext attack.

The following simulation game is played between adversary A and challenger B .

Initialization phase: B runs $\text{Setup}(1^\lambda)$ to produce the system public key $pk = (A_0, g_3, \{A_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq n_i}, Y_0, Y, KF, \omega, \nu, \kappa, H, H_0) \neq$ and the system master private key $msk = (g_1, \{x_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq n_i}, \beta_0, \beta)$. Then, B sends pk to A and generates an initially empty list L and an empty set \mathbb{R} .

Inquiry phase 1: A can initiate the following two types of inquiries to B .

- (1) Private key inquiry: A adaptively asks B for the private key of the attribute set Λ , B runs $\text{Key Gen}(pk, msk, \Lambda)$ and returns $sk_\Lambda = (K_0, K, \{K_{i,j}\}_{1 \leq i \leq k})$ to A . B calculates $\Lambda \cap \mathbb{R}$ and assigns $\Lambda \cap \mathbb{R}$ to \mathbb{R} .
- (2) Transformation key inquiry: receiving the request of token inquiry from A , B first searches for $(\Lambda, sk_\Lambda, tk_\Lambda)$ in list L . If $(\Lambda, sk_\Lambda, tk_\Lambda)$ exists, then B returns tk_Λ to A ; otherwise, B chooses a random number $y \in Z_N$ and calculates $tk_\Lambda = (T_i^*, T^*, T_0^*, I_\alpha^*)$. Then, B adds Λ and tk_Λ to list L and returns list L to A .

Challenge phase: A sends equal-length messages M_0, M_1 and access structure W_0, W_1 to B . B selects $\gamma \in \{0, 1\}$ and runs $\text{Encrypt}(pk, m, W)$ to generate challenge ciphertext $C_T = (C_0 = A_0^s Q_0', C_1 = mY^s, \{C_\alpha\}_{\forall \alpha})$. Finally, B sends C_T to A .

Inquiry phase 2: similar to inquiry phase 1, but A cannot ask for messages M_0 and M_1 .

Guess: A outputs the guess $\gamma' \in \{0, 1\}$. If $\gamma' = \gamma$, then the attack is declared successful. Based on the proof of Definition 5 in Scheme [30], it is difficult for A to guess γ' and γ selected randomly during the ciphertext generation phase. We prove that the confidentiality of our scheme satisfies security under a chosen-ciphertext attack. \square

6.2. Privacy Policy. The DO uploads the ciphertext components $C_0 = A_0^s Q_0', C_1 = mY^s$ and $C_\alpha = (\bar{C}_\alpha, I_\alpha, \{C_{i,j}\}_{(1 \leq i \leq n, 1 \leq j \leq n_i)})$ to the CS, where $C_{i,j} = A_{i,j}^{s_\alpha} Q'_{i,j}$ or $C_{i,j} = A_{i,j}^{s_{i,j}} Q'_{i,j}$. Note that the attribute information s_α is hidden in the ciphertext component C_α . When an attribute value of the accessing user satisfies the value under node α , then the ciphertext component can be obtained by $C_{i,j} = A_{i,j}^{s_\alpha} Q'_{i,j}$, where s_α is the attribute information. When a data user does not meet the access control, the DO uses a random value $s_{i,j}$ to replace s_α and obtains the ciphertext component $C_{i,j} = A_{i,j}^{s_{i,j}} Q'_{i,j}$, even if the data user who does not meet the access control obtains the ciphertext and calculates

$$\begin{aligned}
\omega_\chi &= \frac{I_\alpha^*}{e(C_\alpha, T^*) \text{Pre Dec Node}(\alpha)} = \frac{Y^{s_\alpha/y}}{e(A_0^{s_\alpha} Q_\alpha, g_1^{\beta - \lambda/y}) e(A_{i,j}^{s_{i,j}} Q_{i,j}', K_i^{1/y})^\lambda} \\
&= \frac{e(g_1, g_1)^{\beta s_\alpha/y}}{e(g_1, g_1)^{s_\alpha(\beta - \lambda)/y} e(g_1, g_1)^{\lambda s_{i,j}/y}}.
\end{aligned} \tag{5}$$

There are random values $s_{i,j}$ in the above equation; therefore, users who do not satisfy the access control do not obtain the attribute values of node α . Thus, the whole access structure cannot be inferred from the access policy. Therefore, the scheme in this article satisfies policy privacy.

TABLE 1: Functional comparison.

System	Access structure	Hidden policy	Predetected decryption	Verifiable outsourcing	Integrity	Confidentiality	Blockchain technology
[36]	LSSS	√	×	×	×	CPA	×
[30]	Access tree	√	√	√	×	CPA	×
[34]	LSSS	√	×	√	√	—	√
[29]	LSSS	√	×	√	×	CPA	×
Ours	Access tree	√	√	√	√	CCA	√

6.3. Verifiability

Theorem 4. *For a composite-order bilinear group, if the discrete logarithm problem holds in the system, then the proposed scheme satisfies verifiability.*

Proof. If within the PPT time, the verifiability of the system can be attacked by attacker A with a nonnegligible advantage, then algorithm \mathbb{S} can be simulated to solve the discrete logarithm problem in a composite-order bilinear group system. The bilinear system $(N, p_1, G_0, G_T, \hat{e}, g_1, \Delta = g_1^x)$ is input into the simulation algorithm \mathbb{S} . The algorithm \mathbb{S} needs to calculate $x = \log_{g_1} \Delta$. The game process between the simulation algorithm \mathbb{S} and attacker A is as follows:

Initialization phrase: the simulation algorithm \mathbb{S} randomly generates the parameters $\gamma \in Z_N$, picks two anticollision hash functions, $H: \{0, 1\}^* \rightarrow Z_N$ and $H_0: G_0 \rightarrow Z_N^*$, and defines a key distribution function KF . Later, \mathbb{S} generates system public parameters $pk = (A_0, g_3, \{A_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq n_i}, Y_0, Y, KF, \omega, \nu, \kappa, H, H_0)$ according to the scheme initialization process and sends the public key to attacker A .

Challenge phrase: Attacker A sends the attribute set Λ to the simulation algorithm \mathbb{S} , performs the key generation process $\text{Key Gen}(pk, \text{msk}, \Lambda)$ to generate the private key $sk_\Lambda = (K_0, K, \{K_i\}_{1 \leq i \leq k})$ corresponding to the attribute set Λ and sends it to attacker A .

Output phrase: Attacker A outputs a tuple $(C_T', tk, C_{k_1}, C_{k_2}, \Delta)$ and an encrypted access structure W that satisfies the attribute set Λ . The simulation algorithm \mathbb{S} calculates $KF(nk_1, \kappa) = vk_1 \parallel \varepsilon_1$ and $KF(nk_2, \kappa) = vk_2 \parallel \varepsilon_2$, where $nk_1 = w_{1,2}^\Delta$ and $nk_2 = w_{2,2}^\Delta$. If $nk_1 \neq nk_2$, that is, attacker A wins the game, and the simulation algorithm \mathbb{S} calculates

$$\begin{aligned}
g_1^{x \cdot H(vk_1) + \gamma \cdot H(\varepsilon_1)} &= \omega^{H(vk_1) \nu^H(\varepsilon_1)} \\
&= w_1 \\
&= g_1^{x \cdot H(vk_2) + \gamma \cdot H(\varepsilon_2)} \\
&= \omega^{H(vk_2) \nu^H(\varepsilon_2)}.
\end{aligned} \tag{6}$$

Because the selected hash function H has collision resistance, $vk_1 \neq vk_2$ and $H(vk_1) \neq H(vk_2)$, the algorithm \mathbb{S} is able to compute $x = \gamma(H(\varepsilon_1) - H(\varepsilon_2)) / (H(vk_1) - H(vk_2))$ as

a solution to the discrete logarithm problem, which proves that the proposed scheme is verifiable. \square

6.4. Data Integrity. Data integrity is guaranteed by two processes. First, the smart contract is used to realize the decryption correctness of the outsourcing server. Subsequently, the original data hash on the blockchain is saved to verify the data integrity. After receiving the semidecrypted ciphertext $C_T' = N$ sent by the outsourcing server, the data access user uses the blinding factor γ to calculate the session key $nk = N^{-\gamma} = e(g_1, g_1)^{\beta \gamma}$ and replaces the key allocation function $KF(nk, \kappa) = vk \parallel \gamma$. A smart contract verifies equation $\omega^{H(vk) \nu^H(\gamma)} = \tilde{C}$ and outputs $\text{bool} = 1$ when this equation is established. Then, the data access user continues to decrypt semidecrypted ciphertext. Otherwise, the smart contract outputs $\text{bool} = 0$ and ends the decryption.

After the DAU performs decryption to obtain plaintext m , $\tilde{C}' = \nu^{H_0(m)}$ is calculated and the validity of $\tilde{C}' = \tilde{C}$ is verified. If the equation does not hold, then it cannot be verified by data integrity.

7. Performance Analysis

7.1. Property Analysis. In this section, the functionality of our system is compared with schemes in [29, 30, 34, 36], and the comparison outcomes are shown in Table 1. We can note from Table 1 that our scheme is the only one that meets the requirements of policy hiding, verifiable outsourcing, and data integrity under CCA. Schemes in [29, 34, 36] use outsourcing for decryption operations, but their decryption operations are not very efficient. Moreover, the scheme in [36] does not support the validation of outsourcing decryption results. In addition, schemes in [29, 30, 36] achieve data integrity verification by relying on a trusted cloud server. As a result, the proposed new scheme is able to provide both higher security and fuller functionality than existing similar schemes.

7.2. Performance Evaluation. We compare our scheme with Systems [30, 34, 36], which also use bilinear groups of composite order. The computational cost of these schemes is analysed through three stages: encryption, decryption, and outsourcing decryption, and the comparison results are shown in Table 2. Our scheme mainly considers pair operations and exponential operations in groups G and G_T . We use G and G_T to denote the time to perform an exponential

TABLE 2: Computational overhead comparison.

System	Encryption	Outsourcing decryption Predetected	User decryption (s)	Decryption
[30]	$(1 + n_a + n_a n)G + (1 + n)G_{T_p}$	$(m + n)(T_p + G_T)$	$T_p + G_T$	$2G_T = 0.42$
[34]	$T_p + (3 + 3n_l)G + G_T$	$(2 + n_l)T_p + (1 + 2n_l)G$	0	
[36]	$n_l T_p + (4 + 3n_l)G + G_T$	$n_k T_p + (n_k + 1)G$	$G = 0.72$	
Ours	$(n_a + n_a n)G + nG_T$	$(m + n)(T_p + G_T)$	$T_p + G_T$	$G_T = 0.21$

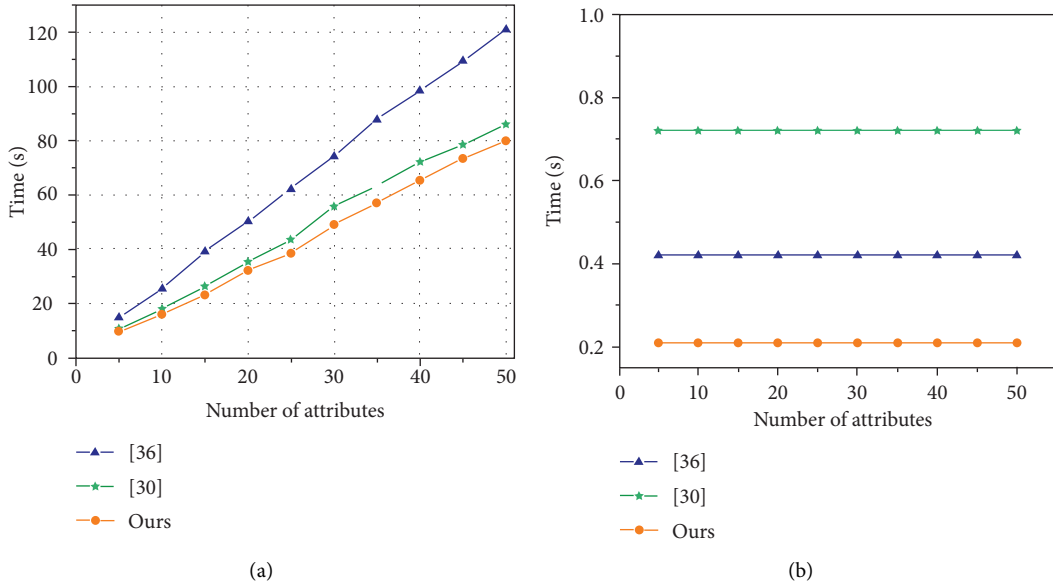


FIGURE 3: Time cost of encryption and decryption with different numbers of attributes. (a) Encryption time of data owner. (b) Decryption time of the user side.

operation on the corresponding group and T_p to denote the time to perform a logarithmic operation. Furthermore, the number of authorized attributes in the system is denoted by n_w , the number of leaf node parents by n_a , the number of attributes in the key by n_k , and the number of user attributes by n_l .

To evaluate the specific computational performance of our scheme, we conducted experiments. Our experimental environment is an Intel(R) Core (TM) i5-8250U CPU 1.80 GHz processor with 8 GB memory and the Win10 operating system (Pairing-Based Cryptography, PBC) library in the VC6.0 environment. Through the above environment, the new scheme was simulated and compared with schemes in [30, 36], and the experimental data were averaged over 20 runs. In the composite-order bilinear group, the times of G , G_T , and T_p are 0.21 s, 0.72 ms, and 1.64 s, respectively. Our scheme and Zhao et al. proposed a scheme in [30] that adopts a special access number structure, and the encryption time is related to the number of parent nodes of leaf nodes n_a . As a result, to better reflect the two systems' performance, we set $n_a = 1$. In addition, we suppose the user has 5 attributes. The number of attributes connected with ciphertext is half the number of systems, and the system contains between 5 and 50 attributes.

In Table 2, we compare these schemes in terms of computational overhead, mainly considering the cost of encryption, outsourcing decryption, and user decryption. For encryption, our scheme improves the efficiency of the ciphertext generation stage. Unlike the scheme in [30], the new scheme uses blockchain technology and minimizes the number of ciphertext components that must be uploaded to the cloud server. Consequently, two exponential operations originally performed by the data owner in the encryption process are reduced. Additionally, in the correctness verification process, the new scheme leaves the verification of the outsourcing results to be performed by smart contracts, reducing the verification overhead for local users. In the decryption phase, all four experiments presented in Table 2 use an outsourced server for predecryption so the decryption overhead for the user is kept at a constant level. The calculation times of the three schemes are G , $2G$, and G_T . Compared with the scheme in [34] without local overhead, and although the new scheme has some decryption overhead, its security is better than the scheme in [34]. On the one hand, when the scheme in [34] uses smart contracts to verify the results of outsourcing, it needs to know the blinding factor that is private for the user. On

the other hand, the smart contract decrypts and obtains the plaintext instead of the user, which makes the plaintext information available to the smart contract and increases the risk of data leakage.

Figure 3 shows the time taken to perform the operation of the data owner and user side. We experiment with different attribute values and show the encryption time changes of the new scheme, the scheme in [30], and the scheme in [34], in Figure 3(a). The computational overhead of the new scheme and the scheme in [30] is smaller than that of the scheme in [34], as shown in Figure 3(a), and the advantage grows as the number of characteristics grows. Due to the additional pair operations and exponential operations in group G that must be computed while hiding the access control policy, the scheme in [34] takes longer. Moreover, based on the scheme in [30], our scheme introduces blockchain technology to encrypt the ciphertext components that need to be encrypted with a data owner in advance in their scheme. This design reduces the encryption time of two exponential operations in the ciphertext generation process.

From Figure 3(b), we can clearly see that the attributes are irrelevant to the time taken for the three schemes to perform decryption (user side) operations, but the time expenditure advantage of our scheme is always higher than those of Schemes [30, 36].

8. Conclusion

We propose a verifiable access control model for outsourced cloud storage that supports policy hiding as well as secure and efficient decryption. Our system is based on the CP-ABE, avoiding privacy leakage by hiding access policies. The idea of outsourcing and a more efficient decryption algorithm reduce the computational cost of local users and outsourcing decryption servers in the decryption process, respectively. To validate the integrity of outsourced decryption results, we use smart contracts implemented on the blockchain, which implements a decentralized ciphertext result verification approach. At the same time, through the hash of the original data retained on the blockchain platform, the integrity of the decrypted data is verified, which solves the dependence of the traditional scheme on fully trusted cloud servers. The analysis results show that the new scheme not only improves computing performance and meets CCA security but also verifies data integrity in the cloud storage environment. In future work, we will attempt to improve the cloud storage data access control scheme for multi-authorization centres.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

All authors have no conflicts of interest.

Acknowledgments

This research was supported by the China Postdoctoral Science Foundation (no. 2017M610817) and the Gansu Science and Technology Planning Project (no. 20CX9ZA076).

References

- [1] H. Yang, Z. Yi, R. Li et al., "Improved Outsourced Provable Data Possession for Secure Cloud Storage," *Security and Communication Networks*, vol. 2021, Article ID 1805615, 12 pages, 2021.
- [2] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-based big data storage systems in cloud computing: perspectives and challenges," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 75–87, 2017.
- [3] W. B. Kim, D. Seo, D. Kim, and I.-Y. Lee, "Group Delegated ID-Based Proxy Reencryption for the Enterprise IoT-Cloud Storage Environment," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 7641389, 12 pages, 2021.
- [4] S. Qi, Y. Lu, W. Wei, and X. Chen, "Efficient data access control with fine-grained data protection in cloud-assisted IIoT," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2886–2899, 2021.
- [5] M. Joshi, K. Joshi, and T. Finin, "Attribute Based Encryption for Secure Access to Cloud Based EHR Systems," in *Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pp. 932–935, San Francisco, CA, USA, July 2018.
- [6] R. Walid, K. P. Joshi, S. Geol Choi, and D.-y. Kim, "Cloud-based Encrypted EHR System with Semantically Rich Access Control and Searchable Encryption," in *Proceedings of the 2020 IEEE International Conference On Big Data (Big Data)*, pp. 4075–4082, Atlanta, GA, USA, December 2020.
- [7] S. Li, Q. Zhang, X. Wu, W. Han, and Z. Tian, "Attribution classification method of APT malware in IoT using machine learning techniques," *Security and Communication Networks*, vol. 2021, Article ID 9396141, 12 pages, 2021.
- [8] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs," *International Journal of Intelligent Systems*, vol. 37, 2021.
- [9] Y. Chen, S. Dong, T. Li, Y. Wang, and H. Zhou, "Dynamic multi-key FHE in asymmetric key setting from LWE," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5239–5249, 2021.
- [10] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pp. 321–334, Berkeley, CA, USA, May 2007.
- [11] B. Waters, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Public Key Cryptography - PKC 2011*, Springer, Berlin, Germany, 2011.
- [12] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008, <http://bitcoin.org/bitcoin.pdf>.
- [13] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, pp. 3596–3612, 2021.
- [14] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, 2021.

- [15] A. Sahai, "Fuzzy identity-based encryption," in *Lecture Notes in Computer Science*, B. Waters, Ed., Springer, Berlin, Germany, pp. 457–473, 2005.
- [16] G. Lin, H. Hong, and Z. Sun, "A collaborative key management protocol in ciphertext policy attribute-based encryption for cloud data sharing," *IEEE Access*, vol. 5, pp. 9464–9475, 2017.
- [17] C. Li, J. He, L. Cheng, C. Guo, and K. Zhou, "Achieving privacy-preserving CP-ABE access control with multi-cloud," in *Proceedings of the 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/Sustain-Com)*, pp. 801–808, Melbourne, Australia, December 2018.
- [18] A. Kapadia, P. P. Tsang, and W. S. Smith, "Attribute-based Publishing with Hidden Credentials and Hidden Policies," in *Proceedings of the Network And Distributed System Security Symposium*, pp. 179–192, NDSS 2007, San Diego, CA, USA, February 2007.
- [19] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based Encryption with Partially Hidden Encryptor-Specified Access Structures," in *Proceedings of the International Conference on Applied Cryptography and Network Security*, pp. 111–129, Springer, Berlin, Heidelberg, June 2008.
- [20] J. Lai, R. H. Deng, and Y. Li, "Fully Secure Ciphertext-Policy Hiding CP-ABE," in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 24–39, Springer, Berlin, Heidelberg, May 2011.
- [21] J. Hur, "Attribute-based secure data sharing with hidden policies in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2171–2180, 2013.
- [22] N. Helil and K. Rahman, "CP-ABE Access Control Scheme for Sensitive Data Set Constraint with Hidden Access Policy and Constraint Policy," *Security and Communication Networks*, vol. 2017, Article ID 2713595, 13 pages, 2017.
- [23] Y. Song, H. Zhen, F. Liu, and L. Liu, "Attribute-based encryption with hidden policies in the access tree," *Journal on Communications*, vol. 36, no. 9, pp. 119–126, 2015.
- [24] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, "Efficient and Provable Secure Ciphertext-Policy Attribute-Based Encryption Schemes," in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 1–12, Springer, Berlin, Heidelberg, April 2009.
- [25] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of abe ciphertexts," in *Proceedings of the USENIX Security Symposium*, San Francisco, CA, USA, August 2011.
- [26] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1343–1354, 2013.
- [27] J. Li, F. Sha, Y. Zhang, X. Huang, and J. Shen, "Verifiable Outsourced Decryption of Attribute-Based Encryption with Constant Ciphertext Length," *Security and Communication Networks*, vol. 2017, Article ID 3596205, 11 pages, 2017.
- [28] J. Zhang, Z. Cheng, X. Cheng, and B. Chen, "OAC-HAS: outsourced access control with hidden access structures in fog-enhanced IoT systems," *Connection Science*, vol. 33, no. 4, pp. 1060–1076, 2021.
- [29] S. Lin, R. Zhang, H. Ma, and M. Wang, "Revisiting attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2119–2130, 2015.
- [30] Y. Zhao, X. Zhang, X. Xie, and S. Kumar, "A verifiable hidden policy CP-ABE with decryption testing scheme and its application in VANET," *Transactions on Emerging Telecommunications Technologies*, p. e3785, 2019.
- [31] D. Di Francesco Maesa, P. Mori, and L. Ricci, "A blockchain based approach for the definition of auditable access control systems," *Computers & Security*, vol. 84, pp. 93–119, 2019.
- [32] Y. Rahulamathavan, R. C.-W. Phan, M. Rajarajan, S. Misra, and A. Kondo, "Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption," in *Proceedings of the 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, Bhubaneswar, India, December 2017.
- [33] Y. Zhang, D. He, and K.-K. R. Choo, "BaDS: blockchain-based architecture for data sharing with ABS and CP-ABE in IoT," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 2783658, 9 pages, 2018.
- [34] F. Zhang, "Research on access control of internet of things based on blockchain and attribute based encryption," Master's Thesis, Nanjing University of Posts and Telecommunications, 2020.
- [35] J. Zhu, K. Hu, and B. Zhang, "Review on formal verification of smart contract," *Acta Electronica Sinica*, vol. 49, no. 4, pp. 792–804, 2021.
- [36] B. Wang and H. Wang, "Research on cloud storage scheme based on attribute encryption," *Journal of Electronics and Information Technology*, vol. 38, no. 11, pp. 2931–2939, 2016.