EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

A Cloud Environment Security Access Control Scheme Based on Federated Learning and Fuzzy Logic Integration

Hongbo Li[1,*]

[1]Information Technology Center, Wenzou Medical University, Wenzhou, Zhejiang, 325035, China

Abstract

In cloud environments, the coexistence of multi-source heterogeneous nodes, cross-domain data sharing, and dynamic access control requirements are mutually intertwined. The lack of capability to address dynamic node risks and differentiated access demands necessitates a solution to these challenges. To this end, a cloud environment security access control scheme integrating federated learning and fuzzy logic is proposed. Firstly, using fuzzy logic to quantitatively evaluate the multidimensional dynamic attributes of nodes in the cloud environment, the results serve as a prerequisite for selecting participating nodes in federated learning; Secondly, a blockchain based federated learning architecture is constructed, and a ciphertext policy attribute based encryption algorithm is introduced to deeply couple access control policies with the federated learning process, achieving fine-grained control where only authorized nodes can participate in model aggregation and decryption. Experimental results demonstrate that this control scheme effectively evaluates the security state of the cloud environment, identifies and defends against multiple attack behaviours, achieves precise permission control for users of varying identities, and ensures the security, reliability, and dynamic adaptability of access control within the cloud environment.

Keywords: federated learning, blockchain, fuzzy logic, cloud environment, access control, cloud environment security status.

Received on 28 January 2026, accepted on 10 April 2026, published on 29 April 2026

doi: 10.4108/eetsis.11731

1. Introduction

With the proliferation of cloud technology in cross-organizational data collaboration scenarios [1], the tension between resource sharing and security safeguards in cloud environments has become increasingly pronounced. On one hand, multiple entities—such as universities and institutions—rely on cloud platforms to facilitate efficient data and resource circulation, thereby dismantling information silos and enhancing collaborative efficiency[2]. On the other hand, the open and distributed nature of cloud environments exposes them to security risks including identity spoofing, unauthorized access, and data breaches[3]. Traditional "one-size-fits-all" access control policies [4] struggle to accommodate the differentiated permission requirements of diverse users and fail to precisely counter dynamically evolving attack behaviors. Significant disparities exist in access demands for shared resources among users of varying identities. Existing access control solutions often hinder legitimate users' normal collaboration due to rigid permission divisions or allow malicious attacks to exploit vulnerabilities through delayed risk identification [5]. Concurrently, the dynamic flow of data within cloud

*Corresponding author. Email: wzlhb168@163.com

environments and the unpredictability of user behaviour [6] further compound the complexity of security governance. Static permission rules alone struggle to cover dynamic scenarios such as “shifting data sensitivity” or “user behavioural anomalies”, frequently resulting in either “over-permissioning” or “under-permissioning” issues [7]. Against this backdrop, establishing a security access control solution tailored to cloud characteristics [8] has become a core requirement for ensuring both the security and efficiency of cross-entity data sharing. This solution must achieve three objectives: firstly, accurately identifying the matching relationship between user identities and permissions to guarantee reasonable access for different roles; secondly, dynamically perceiving risk characteristics during access processes and implementing tiered controls for anomalous behaviour; Thirdly, it must balance security protection with collaborative efficiency, fortifying data security boundaries without impeding normal resource circulation. This requirement has driven the evolution of research from traditional static permission management toward more flexible and intelligent dynamic access control approaches.

Numerous scholars have explored access control solutions. For instance, Karim et al. [9] proposed a hybrid media access control strategy: During non-emergency phases, a priority-based static allocation mechanism operates according to sensor device type and data criticality, ensuring orderly transmission of non-essential data. In emergency phases, a dynamic scheduling method activated based on historical communication records and the urgency of health data prioritizes transmission of critical alerts. A Markov chain model optimizes resource allocation to prevent channel congestion, thereby achieving efficient and orderly access control for sensor devices. However, the static priority mechanism employed during non-emergency phases fails to adequately account for dynamic variations in device communication loads. This may lead to persistent channel resource unavailability for low-priority devices and cumulative data transmission delays, thereby compromising the fairness and timeliness of access control. Raj et al. [10] proposed a blockchain-based access control framework for IoT healthcare systems. By optimizing the attribute-based cryptography algorithm using elliptic curve cryptography, they embedded medical data access policies within ciphertext, binding user private keys to attributes. This ensures only users with matching attributes can decrypt data. Concurrently, Ethereum smart contracts facilitate distributed user authentication, dynamically adjusting access permissions according to clinical contexts. This approach reduces authentication and communication overhead for IoT devices while safeguarding secure access to sensitive medical data. However, Ethereum smart contracts prove inflexible once deployed, hindering timely updates to contract logic when medical access policies require adjustment to evolving operational demands. This inflexibility risks disconnecting access control strategies from practical clinical scenarios. Singh et al. [11] proposed a metapolicy-based access control framework compatible with RBAC (Role-Based Access Control), ABAC (Attribute-Based Access Control), and their combined strategies. This framework encompasses a comprehensive management system covering administrative relationships, operational commands, and contextual constraints. Furthermore, it employs Datalog fact modelling and Z-tools for policy analysis and standardization, unifying the configuration and execution processes of heterogeneous policies. However, because the MPBAC framework requires coordinating adaptation logic for multiple heterogeneous policies, practical implementation often encounters complex policy conflict resolution and diminished access control decision efficiency. Lauer et al. [12], addressing wireless body area networks (WBANs), designed access control logic tailored to device characteristics for data collection and forwarding in wearable/implantable sensors, integrating this into WBAN security frameworks. By regulating access permissions for entities such as sensor nodes and forwarding nodes, they prevented unauthorized data acquisition, thereby enabling controlled access to medical data. However, this approach fails to adequately account for dynamic network topology changes caused by sensor node mobility within WBANs. This can lead to issues such as failed authentication for mobile nodes and unauthorized data forwarding, ultimately compromising the access control mechanism.

Recently, the application research of federated learning in the field of cloud security has been continuously deepening, such as the noise data attack and defense mechanism in federated learning, and the chain continuous learning framework based on quantum federated learning, which provide important references for this research. In the above context, this paper proposes a cloud environment security access control scheme based on the fusion of federated learning and fuzzy logic. By integrating federated learning with fuzzy logic and introducing the CP-ABE algorithm, it implements group-based access control in cloud environments. This provides a viable pathway for fortifying data security boundaries and enabling secure cross-organizational cloud collaboration. The core interaction mechanism of this scheme is as follows: firstly, the fuzzy logic module evaluates the multidimensional dynamic attributes of all nodes in the cloud environment in real time, outputs the security status score of the nodes, and this score serves as the "admission credential" for the selection of participating nodes in federated learning; Then, the federated learning module only allows nodes with scores reaching the threshold to participate in model training and aggregation; Finally, in the federated learning process, the CP-ABE algorithm ensures that only nodes that contribute to the local model and have matching attributes can decrypt the global model. Through this "evaluation screening encryption" cascade mechanism, on-demand and secure access control in cloud environments has been achieved.

2. Cloud Environment Security Access Control

2.1. Fuzzy Logic-Based Dynamic Security Assessment of Cloud Environment Nodes

2.1.1 Fuzzy Logic Factor Influence Values

The security status of cloud environments is seldom a simple dichotomy of secure or insecure, but rather a continuous, gradual process influenced by multiple factors such as network traffic anomaly levels, user behaviour credibility, and system load rates. These factors themselves may prove difficult to quantify precisely (e.g., "behavioral suspicion levels"). Fuzzy logic systems can unify security metrics from diverse sources and with varying dimensions into standardized, semantically clear fuzzy values, thereby better quantifying the state of cloud environments. Consequently, this paper employs fuzzy logic to evaluate node states within the cloud environment prior to implementing access control [13]. This assessment determines the overall state of the cloud environment based on node attributes such as device condition, trustworthiness, and connection stability [14], thereby providing an interpretable basis for subsequent federated learning decisions.

(1) Device Condition Factor: Computational capability is measured by the CPU cycle frequency of cloud environment nodes. Normalizing this frequency yields the device condition influence factor CF_a. The normalization formula adopts the minimum maximum scaling method to map the node CPU frequency to the [0,1] interval. Higher values indicate stronger computational capability, calculated as:

$$CF_a = \frac{f_a - \min\limits_{b\in N}\{f_a\}}{\max\limits_{b\in N}\{f_a\} - \min\limits_{b\in N}\{f_a\}} \tag{1}$$

Where, f_a is the CPU frequency of cloud environment node a; CF_a is the computational capability of node a; min and max are the minimum and maximum CPU frequencies of all nodes, respectively.

(2) Node Trustworthiness Influence Factor: R_{ij} denotes the trustworthiness of nodes within the cloud environment, serving as a measure of their security. Higher trustworthiness indicates stronger current node security. The formula for R_{ij} is:

$$R_{ij} = \frac{N_{ij} \times Q}{\sum_{j=1}^{n} N_{ij}} \tag{2}$$

Where, N_{ij} is the weight of the trustworthiness metric for cloud environment nodes; Q is the maximum trustworthiness weight; n is the number of cloud environment nodes.

(3) Node Stability Impact Factor: $Z-score$ indicates node stability, where a higher value indicates poorer stability. The stability impact factor value for cloud environment nodes is measured based on Z-score detection results, calculated as:

$$Z-score = \frac{X-\mu}{\phi} \tag{3}$$

Where, X is the value of a single data point; μ is the mean of the data; ϕ is the standard deviation of the data.

2.1.2 Dynamic Node Security Assessment

Cloud security metrics are diverse in nature, and the three metrics calculated above exhibit significant differences in their units and ranges. To enable comparability between different metrics, standardization is performed using membership functions, i.e., converting metrics into fuzzy data with values in the [0,1] interval [15]. This maps all metrics onto a unified membership space, facilitating the description of the relative levels of node security status evaluation in the cloud environment. The Gaussian membership function is employed herein, defined by the following formula:

$$gauss(x) = e^{-\frac{(x-\eta)^2}{2\delta^2}} \tag{4}$$

In the formula, η and δ are parameters adjusted based on the tolerance levels of influencing factors to determine the values of η and δ. x represents the specific indicator of the fuzzy parameter, whose fuzzy result is calculated using membership functions. Among them, η determines the center position of the membership function, and δ determines the width of the function. This article sets the η value based on the tolerance threshold of each influencing factor, and sets the δ value based on the fluctuation range of the indicators to control the smoothness of the membership curve.

After determining the membership function, the outputs from Equations (1), (2), and (3) are fuzzified. Fuzzy rules must then be constructed to define the relationship between inputs and outputs, as shown in Table 1.

Table 1. Fuzzy rules

Stability	Credibility	Equipment condition level		
		Poor	Medium	Good
Unstable	poor	poor	poor	poor
Unstable	Medium good	poor	poor	Medium
Unstable	poor	poor	Medium good	good
Medium	Medium good	Medium Medium good	good	good
Medium	poor	good	optimal	optimal
Medium	Medium good	good	optimal	poor

Stable	poor	optimal	poor	Medium
Stable	Medium good	poor	poor	Optimal
Stable	poor	poor	Medium good	Optimal

Inference based on fuzzy rules yields fuzzy outputs, which are then converted back to numerical format using the centroid defuzzification method [16]. The centroid defuzzification formula is:

$$y = \frac{\sum_{i=1}^{n} xgauss(x)}{\sum_{i=1}^{n} gauss(x)} \quad (5)$$

Where, y represents the defuzzified precise value.

After defuzzification, precise values within the range [0-1] are obtained to represent the security level of cloud environment nodes. This enables decision-making for node selection, applying different handling methods—such as isolation, warnings, or access restrictions—based on varying risk levels. This provides a reliable foundation for subsequent cloud environment security access control based on federated learning models.

2.2. Blockchain-Based Federated Learning for Cloud Environment Security Access Control

2.2.1 Blockchain Federated Learning Architecture

Following the precise screening of secure nodes via fuzzy logic, this approach selects trusted nodes for secure data access within cloud environments. Centred on blockchain federated learning methodologies and integrated with the CP-ABE (Ciphertext Policy Attribute-Based Encryption) algorithm, it constructs a secure access control system based on blockchain federated learning. This rigorously governs nodes' data access permissions and operational scope, ensuring that only authorized cloud environment nodes may participate in data sharing, thereby achieving secure data access control in cloud environments.

The essence of federated learning lies in multiple parties jointly participating in model training [17]. It addresses data silos by enabling data providers to share models within a federation without transferring data beyond their local domains [18]. The federated learning framework structure is illustrated in Figure 1. In this scheme, not all participating nodes in federated learning are included, but the security nodes evaluated through fuzzy logic in Section 2.1 are used as admission nodes to construct a trustworthy federated learning environment.

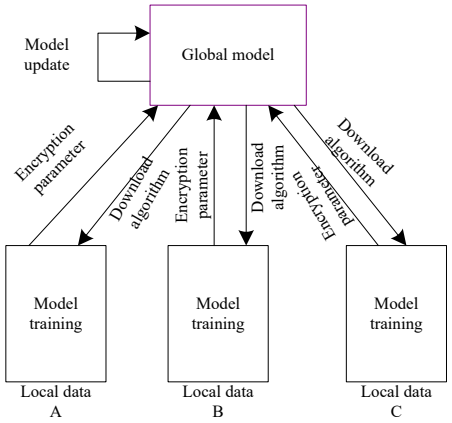

Figure 1. Federated learning structure

Figure 1 is a schematic diagram of the infrastructure of federated learning, showing the interaction between the central server and various participating nodes.

During the initial phase of federated learning, the cloud server provides an initial model to all nodes within the federation. Subsequently, each local node trains this model using its own local data. Upon completion of training, encrypted parameters are transmitted to the cloud server [19]. The cloud server then integrates all node model parameters to form a new global model for use in the next iteration. This article sets the total number of global training iterations for federated learning to E=100, with 5 local training epochs per round, to ensure that the model fully converges. All data in federated learning is trained locally. However, as data sharing is required in the cloud environment, blockchain-based federated learning is employed to ensure secure access control within the cloud infrastructure.

2.2.2 Security Access Control Based on Blockchain Federated Learning

A Blockchain-Based Secure Access Control System for Cloud Environments in Federated Learning [20] first remodels the traditional federated learning workflow [21] through blockchain consensus mechanisms. Decentralized miner nodes replace centralized servers—acting as trusted third parties—to collaboratively complete training and aggregation with local nodes. Building upon this foundation, the system deeply embeds access control mechanisms into the federated workflow [22] to address security requirements for global model access. Each round elects miner nodes to dynamically serve as attribute-encrypted authorities. Access policies are generated based on nodes' historical behaviour within the federated learning process, and the global model is encrypted. Ultimately, only compliant nodes participating in federated learning can obtain decryption keys. This achieves controlled model sharing while safeguarding data privacy.

(1) Blockchain Federated Learning Process

The blockchain federated learning process is illustrated in Figure 2.

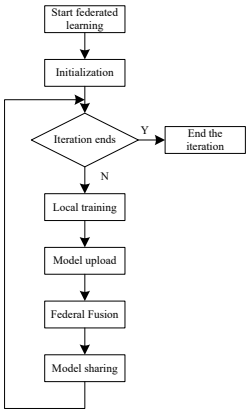

Figure 2. Flowchart of federated learning based on blockchain

Figure 2 is a flowchart of blockchain based federated learning, detailing the five key steps from initialization to model sharing, where miner nodes undertake model aggregation and encryption tasks under the consensus mechanism. The blockchain-based federated learning process comprises five steps:

Step 1: Initialization phase: First, a federated learning initialization request is submitted to the blockchain, embedding the secure node parameters evaluated in the cloud environment as described in Section 2.1, along with the total number of training iterations I_t and the initial model parameters $M_0 = \left(m_0^1, m_0^2, m_0^3, \cdots, m_0^j\right)$, where m_0^j denotes the j-th parameter vector of the 0-th round of the federated learning model. The blockchain then packages these parameters into Block 0 for this task and broadcasts the federated learning task to all federated nodes, laying a reliable foundation for subsequent training.

Step 2: Local training phase: Federated nodes with data receive the broadcast of training information from the blockchain, including the current iteration round i, the current global model M_i, and cloud access control rules. Nodes that do not wish to participate in this iteration ignore the broadcast. Nodes that are willing to participate synchronize the initial model parameters M_i of the current round to their local machines, then check whether the current federated round i is less than the preset total number of iterations I. If it is less, indicating that the federated learning task is not yet completed, they further train their local model on the basis of M_i, obtaining the local model $M_{i+1}^t = \left(m_{i+1}^{t1}, m_{i+1}^{t2}, m_{i+1}^{t3}, \cdots, m_{i+1}^{tj}\right)$, where M_{i+1}^t represents the local model trained by federated node t in the $i+1$-th round of the current federated task, and m_{i+1}^{tj} denotes the j-th parameter vector of m_{i+1}^t.

Step 3: Model upload phase: After completing local training, federated nodes upload their local models integrated with cloud access control to the blockchain. Due to differences in computing power and network conditions among federated nodes, the arrival times of local models at the blockchain may vary. The blockchain will wait for a fixed period of time, discard local models that arrive late, and then enter the federated aggregation phase. At this point, the transaction pool of the blockchain will contain a set of local model $D_{i+1} = \left(m_{i+1}^1, m_{i+1}^2, m_{i+1}^3, \cdots, m_{i+1}^t\right)$ representing the $i+1$-th iteration.

Step 4: Federated Fusion Phase: This phase combines mining—where blockchain nodes compete via computational power to validate and upload model parameters—with federated aggregation. After waiting a fixed duration and collecting sufficient local models in the transaction pool (a temporary buffer where blockchain nodes store access control data), the current round of federated fusion commences.The miner node (the node that wins the competition and is responsible for executing the model aggregation for this round) elects the miner for this round after reaching a proof-of-work consensus. The difficulty value of the proof-of-work consensus algorithm in the system is set very low, so the computational overhead for the miner node is negligible. For each model parameter vector m_{i+1}^{tj}, the miner node calculates the parameter vector of the global model. The formula is:

$$m_{i+1}^{tj} = \sum_{t=1}^{\tau} \frac{m_{i+1}^{tj}}{t} \tag{6}$$

Where, τ represents the number of local models in the transaction pool. Thus, after federated fusion, the miner obtains the global model M_{i+1} for the $i+1$ th round and bundles all participating local models along with the global model for upload to the blockchain. At this point, the iteration count is $i+1$.

Step 5: Model Sharing Phase: After completing a round of federated training, all federated nodes receive a broadcast containing the global model address. Federated nodes synchronize the global model and repeat Steps 2 to 5 based on the global model for the next training round, continuing until the current iteration count i equals the preset total training iterations I_t.

(2) Access Control

During federated learning, all local models and the global model are publicly accessible. This means each federated node can obtain the global model generated for any federated task at any time. However, nodes participating in a specific federated task may not wish to share their training results externally. Therefore, a scheme is required to encrypt the global model [23], allowing only participating federated nodes that contributed models to

decrypt it. Consequently, the CP-ABE algorithm is introduced to implement a fine-grained access control method for federated learning [24].

Traditional CP-ABE comprises four key steps: Step 1: $Setup(\ ) \to TK, MK$: Inputs security parameters γ and outputs global parameters TK and master key MK. Step 2: $ATTRKeyGen(MK, \mathrm{S}) \to SK$: Inputs attribute set S and master key MK, generating corresponding attribute private keys SK for attribute set S. Step 3: $Rncrypt(M, \Delta, TK) \to CT$: Inputs plaintext M, access structure Δ, and public parameters TK, outputting ciphertext CT .Step 4: $Dectypt(TK, CT, SK) \to M$: Input the public parameter TK, ciphertext CT, and the private key SK corresponding to the attribute set S. If the attribute set S satisfies the access structure Δ, decrypt CT and return the plaintext M.

By optimizing the traditional CP-ABE algorithm based on federated learning characteristics [25], this approach replaces the CA (third-party authority center) and data owner roles of traditional CP-ABE with randomly elected miner nodes in each round [26], ensuring the decentralization and security of the scheme. Under this approach, the global model generated by federated learning is accessible only to contributing nodes. Miner nodes achieve one-to-many sharing by encrypting source files just once. The method leverages both symmetric encryption and asymmetric attribute-based encryption. The access control process is illustrated in Figure 3.

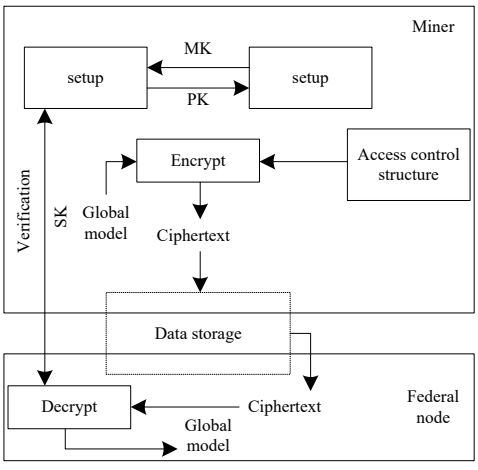

Figure 3. Cloud environment access control process based on blockchain federated learning

Figure 3 is a flowchart of cloud environment access control based on blockchain federated learning, showing how miner nodes encrypt the global model through CP-ABE algorithm and implement fine-grained control flow of decrypting only the contributing nodes in each round of federated learning.

As shown in Figure 3, the proposed method in this section improves upon the traditional CP-ABE approach by adapting it to the federated learning process in a blockchain environment. In decentralized federated learning, it is unsuitable for federated nodes to elect a fixed trusted certificate authority, as a single central point is prone to single-point failures. Therefore, in each round of federated learning, randomly selected miners replace the roles of the trusted certificate authority (CA) and data owner (DO) in traditional CP-ABE. These miners are responsible for encrypting the global model generated in the current round and distributing node private keys. Additionally, due to the simplicity of the encryption target and attribute set, this method is more suitable for federated learning scenarios compared to traditional CP-ABE during the encryption and key generation phases. The workflow of this method is described in detail below in stages:

Step 1: Initialization Phase $Setup(\ ) \to PK, MK, H(M), Key$ (PK : public key; MK : master key; $H(M)$: hash address; Key : decryption key):

Define U_0 as a bilinear group with prime order p, where u is a generator of U_0. A bilinear map $e: U_0 \times U_0 \to U_1$ exists. By randomly selecting two exponents $\alpha, \beta \in Z_p$, the public key PK and master key MK for this round of federated learning are obtained via the formula:

$$PK = \left\{ U_0, u, u^{\beta}, u^{\frac{1}{\beta}}, e(u,u)^{\alpha} \right\} \tag{7}$$

$$MK = \left\{ \beta, u^{\alpha} \right\} \tag{8}$$

The public key PK and master key MK are generated by the miner elected in this round of federated learning via the blockchain's consensus mechanism (a trusted third party) and stored locally on the miner's device. Subsequently, the miner first performs symmetric encryption on the global model to be encrypted using DES (Data Encryption Standard), yielding the encrypted ciphertext M and the decryption key Key. The reason for performing symmetric encryption first is that symmetric encryption algorithms are faster than attribute-based encryption. Therefore, the global model undergoes symmetric encryption to minimize performance overhead. Subsequently, the ciphertext M is uploaded to IPFS (InterPlanetary File System, i.e., the cloud data storage location) and returned as a ciphertext M with an IPFS hash address $H(M)$. This hash address is then uploaded to within the block produced by this miner for public

viewing. Finally, the miner inputs the decryption key Key as plaintext M for the next encryption phase.

Step 2: Encryption Phase $Encrypt(PK, M, T) \rightarrow H(CT)$:

The input parameters for the encryption phase are the public key PK, the plaintext M, and the access construction tree T. In T, honest nodes refer to those whose uploaded local models are accepted by the federated learning Byzantine fault-tolerant algorithm; federated nodes denote those participating in the current federated learning task; miner nodes refer to miners elected in any previous round before this federated task. The encryption algorithm constructs a polynomial q_x for each node in the tree T, then randomly selects a secret value $s \in Z_p$. Let Y represent each leaf node in the tree T. The ciphertext CT can be expressed as:

$$CT = \left\{ T, Me(u,u)^{\alpha s}, h^s, \forall y \in Y : u^{q_y(0)}, H(att(y))^{q_y(0)} \right\} \quad (9)$$

Where, $att(y)$ is the attribute corresponding to leaf node y; $\forall y \in Y : u^{q_y(0)}$ is the encrypted item for each leaf node y in the access tree T; and h^s is the encrypted item of hash function h in the public key PK.

Finally, the miner uploads the ciphertext CT to IPFS. Upon receiving the IPFS address $H(CT)$ in response, it stores this address in the block for public viewing.

Step 3: Key Generation Phase $KeyGen(MK, ADDR) \rightarrow SK$:

During the key generation phase, each federated node seeking decryption eligibility sends a request to the miner node using the block address $ADDR$ signed with its own private key.The cryptographic properties of the blockchain ensure the miner node can verify the authenticity of $ADDR$. The miner node then validates the attributes of $ADDR$ for this federation task using historical block information and assigns it the attribute A. In the system's default environment, A can be an honest node, a federation node, or a miner node. Subsequently, the miner node randomly selects o and $o_j \in Z_p$, calculates the private key SK for this address, and returns it to the requesting node. The private key generation formula is:

$$SK = \left\{ u^{\frac{\alpha+o}{\beta}}, \forall j \in A : u^o \cdot H(j)^{o_j}, u^{o_j} \right\} \quad (10)$$

Step 4: Decryption Phase $Encrypt(H(CT), SK)$:

To obtain the global model, the federated node retrieves the encrypted hash addresses $H(CT)$ and $H(M)$ from block information. It then synchronizes the specific encrypted data CT and the global model encryption M from the IPFS network. Using the requested SK, it performs CP-ABE decryption. Based on the accessing node's attributes, it obtains the corresponding authorized decryption key Key for the global model. This key is used to decrypt M_0, ultimately yielding the global model matching the accessing node's attributes, thereby completing secure access control in the cloud environment.

3. Experimental Analysis

3.1. Experimental Subjects

To validate the effectiveness of this method, an experiment was conducted using a university consortium comprising eight institutions in a certain city. These eight universities decided to adopt blockchain technology to build a cross-campus data sharing platform, aiming to break down inter-institutional information barriers, accelerate the circulation of high-quality educational resources, and promote the joint cultivation of multidisciplinary talents.The platform integrates core data such as course resources, academic achievements, practical training base information, and faculty strength from each institution. Leveraging blockchain's distributed storage and immutability features, it ensures data authenticity and security during sharing.Simultaneously, the platform implements a granular access control scheme. Based on user identity (e.g., enrolled students/faculty, researchers, management personnel) and permission levels, it precisely limits the scope of data that can be accessed and utilized. This approach facilitates efficient inter-institutional collaboration while establishing robust safeguards for data privacy and intellectual property protection. The blockchain topology for data sharing among participating universities is illustrated in Figure 4. Parameters for shared data are detailed in Table 2.

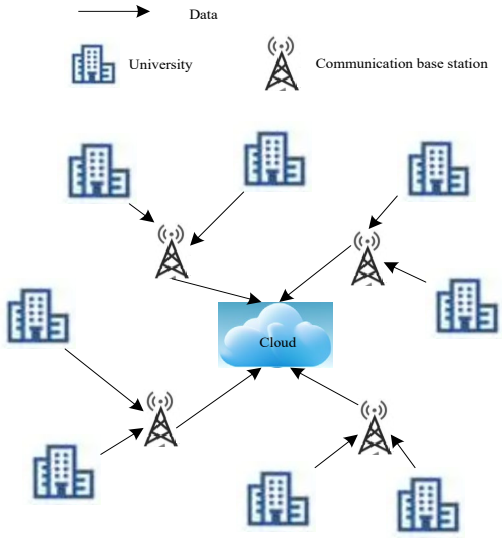

Figure 4. Topological structure of data sharing in Universities

Table 2. Data sharing and access control parameters

Serial number	Parameter category	Parameter name	Parameter value
1	Infrastructure parameters	The number of participating nodes	32 (8 universities, each with 4 independent nodes)
2	Infrastructure parameters	Block generation time	30 seconds per block, balancing data synchronization efficiency and storage pressure
3	Data resource parameters	Shared data type	Course resources, academic achievements, information on practical training bases, faculty resumes, scientific research equipment ledgers, joint enrollment data, etc
4	Data resource parameters	Data update frequency	Support real-time updates (such as academic achievements) and scheduled updates (such as course schedules, once a day in the early morning)
5	Operation and maintenance management parameters	Data backup strategy	Full node local backup + cross-node off-site backup, with a backup cycle of 24 hours
6	Operation and maintenance management parameters	Log retention duration	Operation logs and access logs are retained for 10 years to meet the requirements of educational data compliance audits
7	Performance index parameters	Peak concurrent access volume	Supports 5,000 users to access and call data simultaneously online
8	Cloud server parameters	Cloud server deployment mode	Hybrid cloud deployment (private cloud for core nodes and public cloud for ordinary access nodes)
9	Cloud server parameters	Maximum number of concurrent connections	Supports 10,000 concurrent user connections to ensure stable access during peak hours
10	Cloud storage parameters	Cold/hot data storage strategy	Hot data (accessed within the last 3 months) is stored on SSD, while cold data (accessed over 3 months) is stored on low-cost object storage
11	Cloud storage parameters	Data disaster recovery and backup mechanism	Cross-regional three-copy backup, with RTO of backup data ≤1 hour and RPO≤5 minutes
12	Cloud network parameters	Data Transmission Protocol	HTTPS+ dedicated line encrypted transmission, supporting IPv6/IPv4 dual-stack networks
13	Cloud network parameters	Network latency threshold	The average data synchronization delay between inter-school nodes is ≤50ms, and the average user access delay is ≤100ms
14	Cloud platform operation and maintenance parameters	Platform monitoring methods	7× 24-hour full-link monitoring, covering dimensions such as node status, data transmission, and user access

The parameter configuration in Table 2 is based on the actual needs of the university alliance: there are 32 nodes, and each university deploys 4 independent nodes, covering four functional domains: administration, academic affairs, scientific research, and resources; The block generation time is set to 30 seconds, balancing data synchronization efficiency and storage pressure; The data backup cycle is 24 hours to ensure that the data recovery point objective (RPO) meets educational data compliance requirements.

3.2. Experimental Results

The experimental environment is uniformly configured with CPU Intel Xeon Gold 6248, 128GB of memory, operating system Ubuntu 20.04, blockchain underlying using Hyperledger Fabric 2.5, and federated learning framework based on FATE v1.10. The test duration is 30 minutes, and the average performance of the system during stable operation is recorded.

To validate the efficacy of fuzzy logic in assessing node security in cloud environments under this methodology,

attacks of varying risk levels were conducted against the institution's cloud nodes (low-risk: unauthorized account access; medium-risk: weak password brute-force attacks; high-risk: database compromises; extreme-risk: malicious privilege escalation). The fuzzy logic-based security assessment outcomes for these attacks are illustrated in Figure 5.

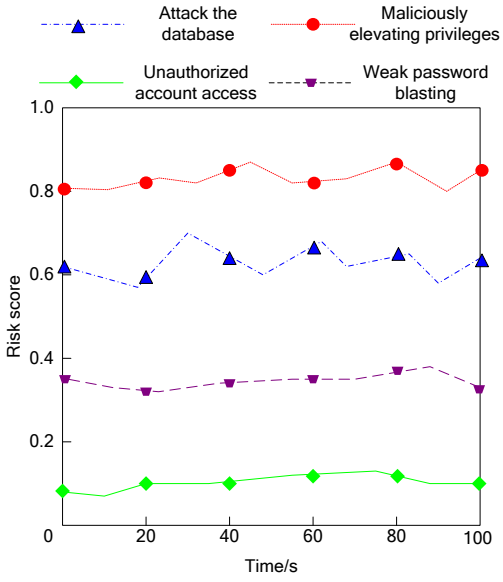

Figure 5. Safety assessment effect

The risk scoring results in Figure 5 demonstrate that the fuzzy logic proposed in this paper accurately matches the risk levels of different attacks in the cloud environment: for attacks of varying risk grades, the risk scores output by fuzzy logic exhibit clear hierarchical distinctions. This outcome stems from fuzzy logic's multidimensional quantification of attack behaviour characteristics: precise risk level mapping is achieved through the calculation of membership functions for fuzzy indicators such as 'node stability, node trustworthiness, and device condition factors'. Compared to traditional binary judgements, the fuzzy logic's graded scoring system not only reflects the differential harm of attacks but also aligns closely with their actual risk levels. This demonstrates its effectiveness in identifying the risk tiers of attacks targeting chain nodes within cloud environments, providing precise grounds for selecting access control nodes.

To validate the security access control performance of this methodology, attacks were launched against selected nodes during university data access processes. Compromised nodes were then used to access cloud data. The cloud data access outcomes under various attack methods are presented in Table 3.

Table 3. Data access situations of different attack methods

Serial number	The number of the attacked node	Attack method	Read the amount of data (MB)	Fuzzy logic risk scoring	Decision-making result
1	University 2-Node 3	Weak password blasting	12.5	0.78 (High Risk)	Warning
2	University 5-node 2	Man-in-the-middle attack	8.3	0.65 (Medium to high risk)	Restricted access
3	University 1-Node 4	SQL injection attack	0	0.82 (High Risk)	Direct interception
4	University 7-Node 1	Privilege escalation attack	25.7	0.52 (Medium risk)	Intercept core data
5	University 3-Node 3	DDoS attack	0	0.91 (Extremely High Risk)	Node fuse
6	University 6-node 4	Malicious code injection	18.9	0.60 (Medium to high risk)	Post-visit traceability + data destruction
7	University 4-Node 2	Forged digital certificate	0	0.75 (High Risk)	Denied access

The results from the seven attack tests in Table 3 demonstrate that this cloud environment access control method, incorporating fuzzy logic, effectively curtails data access attempts by malicious nodes within the federated learning blockchain. For six typical attacks—including weak password brute-force attacks and man-in-the-middle attacks—fuzzy logic calculates multi-dimensional membership scores based on node behaviour, generating risk ratings that trigger differentiated control strategies such as 'alert', 'restrict access', or 'direct interception'. Notably, four attack types (SQL injection, DDoS, certificate forgery, etc.) resulted in zero data readouts, achieving complete data leakage prevention. direct interception". For four attack groups (SQL injection, DDoS, certificate forgery, etc.), the data read volume was zero, achieving zero data leakage. Only two attack types (privilege escalation and malicious code injection) resulted in data readouts, yet these amounted to 25.7MB and 18.9MB respectively, both involving low/medium-sensitivity data (excluding core models or high-value academic resources). This validates the principle that 'even if defences are breached, substantial high-sensitivity data

cannot be obtained,' effectively safeguarding data access security.

To validate the fault-tolerant capability of the proposed method's federated learning blockchain, it was compared with Bulyan, NoDefense, and Krum's approaches. The comparison assessed data access effectiveness under varying numbers of compromised nodes during label flipping attacks (which preserve data features while corrupting samples by tampering with labels, rendering data unusable due to widespread errors). Results are presented in Figure 6.

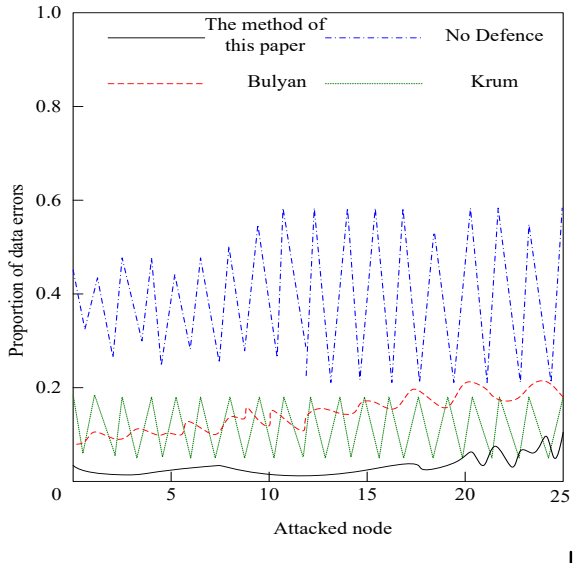

Figure 6. Fault-tolerant capability of access control

The experimental results in Figure 6 demonstrate that during label flipping attacks, as the number of compromised nodes progressively increases from 0 to 25, the 'data error rate' corresponding to the proposed method consistently remains below 0.2 within an extremely low range, exhibiting minimal overall fluctuation and exceptional stability. In contrast, the 'NoDefense' approach consistently exhibited a high error rate exceeding 0.4 with significant fluctuations. While 'Krum' and "Bulyan" outperformed 'NoDefense', their error rates fluctuated more substantially and remained higher than those of the proposed method. This outcome fully validates the robust fault tolerance of our federated learning blockchain approach: leveraging blockchain's distributed trusted architecture and access control mechanisms, it effectively isolates the impact of compromised data from attacked nodes. Combined with precise control over data interactions via access policies, it consistently limits the spread of data errors even as the number of attacked nodes increases, ensuring reliable data access. This characteristic provides critical support for the security of federated learning systems under scenarios such as label flipping attacks.

To validate the effectiveness of this method's security access control in cloud environments, data access was randomly performed by users representing three roles—teachers, administrators, and students—within a selected institution. The access results are presented in Table 4.

Table 4. Results of access control for different attributes

Access identity	File name	Visit result
Teacher	"Cross-school Joint Course Teaching Plan	Successful access. Supports editing, downloading and sharing permissions
	"Shared Ledger of Scientific Research Equipment (Full)	Upon successful access, you can view device details and submit a reservation application
Managers	"Cross-school Joint Course Teaching Plan	Successful access only supports viewing and exporting permissions, no editing rights
	"Shared Ledger of Scientific Research Equipment (Full)	Upon successful access, you can view, edit the device status and review the reservation application
Student	"Cross-school Joint Course Teaching Plan	Successful access. Only online viewing is supported. No download or editing rights are available
	"Shared Ledger of Scientific Research Equipment (Full)	Access is blocked. Only the basic information of the public device (model, location) is allowed to be viewed.

The access control results in Table 4 demonstrate that our method achieves differentiated and stringent access management through precise mapping of identity attributes and permission policies: For the two document types, 'Cross-Institution Joint Course Lesson Plans' and 'Research Equipment Shared Ledger (Full Version)', the system dynamically assigns permissions aligned with role requirements based on the identity attributes of teachers, administrators, and students. Teachers, as the primary collaborators in teaching, are granted editing and downloading permissions for lesson plans, along with reservation application rights for equipment ledgers; Administrators, as resource custodians, possess rights to view and export lesson plans, alongside approval and editing privileges for equipment ledgers. Students, as foundational users, are granted online viewing rights for lesson plans only. Access to the full equipment ledger is blocked, permitting them to retrieve only publicly available

basic information. This outcome validates the core advantage of the proposed methodology: its fuzzy logic-based identity-permission matching mechanism achieves granular permission allocation through 'role adaptation' while ensuring rigid boundary constraints effectively intercept any access attempts exceeding a user's designated identity attributes. This 'authorisation-on-demand plus unauthorised access interception' model safeguards the efficiency of cross-institutional data sharing while fortifying access security boundaries for users of differing identities. It fully demonstrates the precision and reliability of this approach in cloud-based access control.

To comprehensively evaluate the overall performance of our method, we compared it with the blockchain based access control framework proposed in reference [10] and the CP-ABE scheme proposed in reference [24] in the same experimental environment, focusing on two key indicators: access control decision delay and system throughput. The experiment used the same university alliance data sharing platform to simulate 1000 concurrent users initiating data access requests. The average decision delay (ms) of access control policies under different methods and the number of requests processed per second (QPS) of the system were recorded. The results are shown in Table 5.

Table 5. Performance comparison of different access control methods

Method	Average decision delay (ms)	System throughput (QPS)
Reference [10] method	78.6	1250
Reference [24] method	95.2	980
Proposed Method	52.3	1680

From Table 5, it can be seen that the average decision delay of our method is 52.3ms, which is 33.5% and 45.1% lower than the methods in references [10] and [24], respectively; The system throughput reached 1680 QPS, an increase of 34.4% and 71.4% respectively. This is mainly because the method proposed in this article uses fuzzy logic to pre screen trusted nodes, reducing the participation of invalid nodes in the federated learning process. At the same time, it uses miner nodes to replace traditional CAs, avoiding single point performance bottlenecks and significantly improving the real-time and system scalability of access control.

4. Conclusion

This paper investigates a cloud environment security access control scheme based on the integration of federated learning and fuzzy logic. It innovatively combines federated learning with fuzzy logic, employing blockchain federated learning and the CP-ABE algorithm for access control. By incorporating fuzzy logic to evaluate the security status of nodes, it achieves secure access control within cloud environments. Experimental validation confirms this method can implement secure access control in cloud environments, thereby ensuring data security.

From the perspective of computational complexity, the main overhead of this method is focused on fuzzy logic reasoning and the encryption and decryption process of CP-ABE. The complexity of fuzzy logic reasoning is linearly related to the dimensions of input indicators and the number of fuzzy rules. In this paper, the complexity of a single evaluation is O (n), where n is the number of nodes; The encryption complexity of CP-ABE is related to the access tree structure, and the decryption complexity is related to the number of attributes. With the support of cloud server computing power, this overhead is acceptable. From the perspective of scalability, the blockchain based federated learning architecture naturally supports dynamic joining and exiting of nodes, combined with fuzzy logic for security evaluation of newly added nodes, which can effectively maintain system stability. In terms of actual deployment, this method has no special requirements for the underlying hardware and can be integrated with existing cloud platform management systems by deploying smart contracts and federated learning service modules. Future research may explore lightweight deployment of this approach to reduce computational overhead, thereby adapting it to resource-constrained scenarios such as edge computing and IoT terminals, and expanding its application across diverse industries.

References

[1] Sharma N, Saharia M. ML-cascade: a machine learning and cloud computing-based tool for rapid and automated mapping of landslides using earth observation data. Landslides. 2025; 22(1):31-43.
[2] Gazis A, Katsiri E. Streamline intelligent crowd monitoring with IoT cloud computing middleware. Sensors. 2024; 24(11):3643-3665.
[3] Bari MF, Chowdhury MR, Sen S. A computational harmonic detection algorithm to detect data leakage through EM emanation. IEEE Internet Things J. 2025; 12(16):32916-32931.
[4] Jayasundara SH, Arachchilage NAG, Russello G. SoK: access control policy generation from high-level natural language requirements. ACM Comput Surv. 2025; 57(4):1-37.
[5] Kaya TT, Yalcin E, Kaleli C. A novel classification-based shilling attack detection approach for multi-criteria recommender systems. Comput Intell. 2023; 39(3):499-528.
[6] Mousa K, Zhang Z, Sumarliah E, Hamdan IKA. The impact of cloud computing adoption on firm performance among SMEs in Palestine: a machine learning approach. Int J Intell Inf Technol. 2024; 20(20):1-24.
[7] Fuente-Anaya HADL, Marin-Castro HM, Garcia-Hernandez JJ. Business process discovery as a service with event log privacy and access control over discovered models. Computing. 2024; 106(11):3603-3625.
[8] Hmidi Z, Kahloul L, Benharzallah S. A new mobility and energy harvesting aware medium access control (MEH-

MAC) protocol: modelling and performance evaluation. Ad Hoc Netw. 2023; 142(4):103108.
[9] Karim M, Rahman MA, Atiquzzaman M. Hybrid medium access control strategy for Internet-of-Things-enabled intravehicular health monitoring system. IEEE Internet Things J. 2025; 12(4):3846-3857.
[10] Raj A, Prakash S. An efficient blockchain-based access control framework for IoT-healthcare system. Wirel Pers Commun. 2024; 136(2):1017-1045.
[11] Singh MP, Sural S, Vaidya J, Atluri V. A role-based administrative model for administration of heterogeneous access control policies and its security analysis. Inf Syst Front. 2024; 26(6):2255-2272.
[12] Lauer H, Rudolph C, Grobler M, Shahraki AS, Sakzad A. Access control, key management, and trust for emerging wireless body area networks. Sensors. 2023; 23(24):9856.
[13] Otieno SO, Wambua JM, Mwema FM, Mharakurwa ET, Jen TC, Akinlabi ET. A predictive modelling strategy for warpage and shrinkage defects in plastic injection molding using fuzzy logic and pattern search optimization. J Intell Manuf. 2025; 36(3):1835-1859.
[14] Korenevskiy NA, Al-Kasasbeh RT, AshrafShaqadan, Al-Habahbeh OM, AhmadTelfah, Mousa MS, Rodionova S, Filist S, Al-Kassasbeh ET, Krutskikh V, Shalimova E, Aikeyeva AA, Ilyash M. Computerized decision support system and fuzzy logic rules for early diagnosis of pesticide-induced diseases. Crit Rev Biomed Eng. 2025; 53(1):1-22.
[15] Enaya YA, Karim AA, Saleh SM, Shneen SW. Adapting wired TCP for wireless ad-hoc networks using fuzzy logic control. J Eur Syst Autom. 2024; 57(5):1377-1386.
[16] Araz OU, Ilgin MA, Eski O, Araz C. Fuzzy demand-driven material requirements planning: a comprehensive analysis of fuzzy logic implementation in DDMRP. Int J Prod Res. 2024; 62(21):7793-7811.
[17] Nakai-Kasai A, Wadayama T. Regular section deep unfolding-based weighted averaging for federated learning under device and statistical heterogeneous environments. IEICE Trans Commun. 2025; 108(4):411-420.
[18] Dao MC, Nguyen PL, Pham HH, Nguyen TH, Chen P, Wahib M, Truong TN. Noisy data-based attack: a new type of untargeted attack in federated learning and its countermeasures. Future Gener Comput Syst. 2025; 173(12):107900.
[19] Ferretti S, Cassano L, Cialone G, D'Abramo J, Imboccioli F. Decentralized coordination for resilient federated learning: a blockchain-based approach with smart contracts and decentralized storage. Comput Commun. 2025; 236(4):108112.
[20] Alebouyeh Z, Bidgoly AJ. Privacy-preserving federated learning compatible with robust aggregators. Eng Appl Artif Intell. 2025; 143(3):110078.
[21] Gurung D, Pokhrel SR. Chained continuous quantum federated learning framework. Future Gener Comput Syst. 2025; 169(8):107800.
[22] Barbieri L, Kianoush S, Nicoli M, Serio L, Savazzi S. A close look at the communication efficiency and the energy footprints of robust federated learning in industrial IoT. IEEE Internet Things J. 2025; 12(11):15130-15150.
[23] Ferrer-Rojas A, Maharaj BTJ. Multiauthority KP-ABE access model with elliptic curve cryptography. SAIEE Afr Res J. 2025; 116(2):59-67.
[24] Sravanthi K, Chandrasekhar P. An efficient multi-user groupwise integrity CP-ABE (GI-CPABE) for homogeneous and heterogeneous cloud blockchain transactions. J Electr Syst. 2024; 20(1):326-349.
[25] Shruti, Rani S, Boulila W. Securing Internet of Things device data: an ABE approach using fog computing and generative AI. Expert Syst. 2025; 42(2):e13691.
[26] Fugkeaw S, Suksai P, Hak L. SSF-CDW: achieving scalable, secure, and fast OLAP query for encrypted cloud data warehouse. J Cloud Comput. 2024; 13(1):129.