Research on Data Security Protection and Access Control Algorithms in Cloud Computing Environments

Rui Zhang, Yan Gao *

The 30th Research Institute of China Electronics Technology Group Corporation 610041

Abstract. With the rapid development of cloud computing technology, data security and access control have become critical issues in cloud computing environments. This paper aims to research and design an effective data security protection mechanism and access control algorithm to ensure the security and rationality of data access in cloud environments. Firstly, this paper provides an overview of data security issues in cloud computing environments and analyzes the shortcomings of existing access control models. Secondly, a role-based access control (RBAC) model is proposed, combined with the characteristics of attribute-based access control (ABAC), to design a new access control algorithm. This algorithm not only improves the flexibility and fine-grained access control but also enhances data security by introducing encryption technology and access control lists (ACLs). Finally, the effectiveness and security of the proposed algorithm are verified through experiments. The research results show that the algorithm can meet the high requirements of data security and access control in cloud computing environments.

Keywords: cloud computing; data security; access control; RBAC.

1. Introduction

1.1 Development of Cloud Computing and Its Impact on Data Security

Cloud computing, as a new computing paradigm, provides computing resources and services over the internet, with its high efficiency, flexibility, and low cost. The rapid development of cloud computing has greatly promoted the progress of information technology and changed the traditional way of data storage and computing. However, with the widespread adoption of cloud computing, data security issues have become increasingly prominent. Since cloud computing environments rely heavily on third-party service providers for data storage and processing, users' data security and privacy protection face significant challenges. Therefore, researching how to effectively protect data security in cloud computing environments has become a hot topic in academia and industry [1].

1.2 Research Background and Significance

Data security issues in cloud computing environments involve multiple aspects, including data storage security, transmission security, access control, and privacy protection. Traditional security measures are difficult to fully adapt to the characteristics of cloud computing, especially in the face of complex and dynamic network environments and multi-tenant shared resource architectures. Data security faces higher risks, such as data theft, tampering, and unauthorized access. Therefore, researching new data security protection mechanisms and access control algorithms not only has important theoretical significance but also has practical application value in improving data security levels in cloud computing environments.

1.3 Research Objectives and Main Contributions

The research objective of this paper is to propose a new algorithm for data security protection and access control in cloud computing environments. Specifically, the main contributions of this paper include:

(1) Analyzing data security threats in cloud computing environments: providing a systematic analysis of data security threats in cloud computing environments to provide a theoretical basis for algorithm design.

(2) Designing a new access control algorithm: combining RBAC and ABAC models to propose a new comprehensive access control algorithm to improve access control flexibility and security [2,3,4].

(3) Introducing multi-level encryption technology: designing a multi-level encryption mechanism to enhance data storage and transmission security.

(4) Experimental verification and performance analysis: verifying the effectiveness of the proposed algorithm through simulation experiments and analyzing its performance in terms of data security and access efficiency.

2. Current Status of Cloud Computing Data Security

2.1 Data Security Issues

Data security issues in cloud computing environments are mainly reflected in the following aspects:

Data storage security: due to the centralized storage mode of cloud computing, data stored in the cloud faces the risk of unauthorized access and malicious attacks.

Data transmission security: in cloud computing environments, data is frequently transmitted between users and the cloud, making it vulnerable to theft and tampering.

Access control issues: traditional access control mechanisms are difficult to adapt to the dynamic changes and multi-tenant characteristics of cloud computing environments, making it challenging to effectively manage data access permissions.

Privacy protection issues: user data stored in the cloud may be accessed and utilized by cloud service providers or other third parties, making it difficult to guarantee user privacy protection.

The 2019 Capital One data breach incident, where hackers exploited a cloud service configuration vulnerability to access Capital One's database and steal personal information of 100 million American users and 6 million Canadian users, exposed the inadequacies of cloud computing environment configuration management and access control mechanisms. The 2017 Verizon data breach incident, where a third-party supplier misconfigured an Amazon S3 storage bucket, resulting in the exposure of 14 million customers' personal information, further highlights the risks of access control failures in cloud computing environments. Both incidents demonstrate that data security in cloud computing environments not only relies on advanced access control algorithms but also requires strict configuration management and continuous security monitoring.

2.2 Analysis of Existing Access Control Models

Currently, the most commonly used access control models in cloud computing environments are Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC). Both models have their advantages and disadvantages:

Role-Based Access Control (RBAC): RBAC manages user permissions through roles, which has the advantages of being simple and easy to use, and convenient for management [5]. However, in the face of complex and dynamic cloud computing environments, RBAC's flexibility is insufficient, and it is difficult to meet the needs of fine-grained access control.

Attribute-Based Access Control (ABAC): ABAC decides access permissions based on user attributes, resource attributes, and environment attributes, which has higher flexibility and fine-grained control capabilities [6]. However, ABAC's implementation and management are more complex, and the computational overhead is larger.

2.3 Comparison of Role-Based and Attribute-Based Access Control Models

In summary, RBAC and ABAC have their own advantages and disadvantages, and using a single model cannot fully meet the access control needs of cloud computing environments [7]. Therefore, this paper proposes a comprehensive access control algorithm that combines RBAC and ABAC to achieve efficient and flexible access control.

On the basis of user roles, attribute constraints are added to further refine the permissions of roles. For example, a user can be assigned the role of "administrator", but their specific permissions still need to meet specific attribute conditions, such as time and location. By combining the dynamic characteristics of ABAC, the permissions of users can be adjusted in real-time according to changes in environment attributes, ensuring the flexibility and security of access control. To this end, a unified policy language is defined to describe the combination rules of roles and attributes, simplifying policy management and maintenance.

Using a unified policy language to define the combination rules of roles and attributes. For example, the administrator role can access all data during working hours, but only access part of the data during non-working hours. When a user logs in, permissions are assigned based on their role and attributes. First, basic permissions are assigned based on the role, and then refined and adjusted based on attributes. When a user accesses resources, their attributes are evaluated in real-time to determine whether they meet the access policy. If they do, access is allowed; otherwise, access is denied [8].

By combining the advantages of RBAC and ABAC, more flexible access control is achieved. User permissions are not only based on roles but can also be adjusted based on dynamic attributes. Through further refinement of attributes, more precise access control is achieved, improving system security. Additionally, a unified policy language simplifies complex permission management in complex environments, making the definition and management of access rules more efficient.

AWS provides a rich set of security tools and services, such as IAM (Identity and Access Management), Cognito (User Identity Verification), and KMS (Key Management Service), making it relatively easy to implement a comprehensive access control algorithm on the AWS platform. Developers can use these services to quickly build role and attribute management mechanisms and implement dynamic permission evaluation using Lambda functions. The diversity and complexity of AWS services may increase the learning cost for beginners.

Azure's security services, such as Azure AD (Azure Active Directory) and Azure Key Vault, provide similar identity management and encryption functionality. Azure's policy definition language (Azure Policy) can be used to implement complex access control policies. Due to Azure's tight integration with the Microsoft ecosystem, implementing a comprehensive access control algorithm is relatively easier for enterprises using Microsoft technology stacks. However, Azure's service complexity also requires a certain amount of learning and configuration time.

GCP's IAM and Cloud Identity-Aware Proxy (IAP) services can support fine-grained access control. GCP's advantages in big data processing and machine learning can be used to optimize behavior analysis modules. However, GCP's security services are relatively fewer, and more custom development may be needed to meet specific access control requirements. For teams unfamiliar with the GCP environment, the initial implementation difficulty is relatively high.

3. Algorithm Implementation and Analysis

3.1 Algorithm Details

This paper proposes a comprehensive access control algorithm that combines the advantages of Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) to solve the data security problem in cloud computing environments [9,10,11]. The algorithm achieves flexible and fine-grained access control by combining the strengths of RBAC and ABAC.

Permission Assignment: When a user logs in, the system assigns permissions based on their role and attributes. First, basic permissions are assigned based on the role, and then refined and adjusted based on attributes. For example, a user is assigned the "administrator" role, but their specific permissions need to satisfy specific attribute conditions, such as working hours and location.

Real-time Evaluation: When a user accesses resources, the system evaluates their attributes in real-time to determine whether they meet the access policy. If they do, access is allowed; otherwise,

access is denied. The evaluation process includes checking the user's, resource's, and environment's current attributes and matching them with the constraints in the policy.

3.2 Implementation Method

3.2.1 Role Permission Assignment Formula

$$P_{role}(u) = \bigcup_{r \in R_u} permissions(r)$$

where $P_{role}(u)$ represents the set of permissions assigned to a user u based on their role R_u, and $permissions(r)$ represents the set of permissions corresponding to the role r.

3.2.2 Attribute Constraint Formula

$$P_{attr}(u) = \bigcap_{a \in A_u} conditions(a)$$

where $P_{attr}(u)$ represents the set of permissions assigned to a user u based on their attribute A_u, $conditions(a)$ and represents the constraint conditions corresponding to attribute a.

3.2.3 Comprehensive Permission Formula

$$P(u) = P_{role}(u) \cap P_{attr}(u)$$

Where $P(u)$ represents the final comprehensive permission set assigned to a user u.

3.3 Implementation Process

3.3.1 Combining Roles and Attributes

Role Foundation: First, basic permissions are assigned based on the user's role. Role definition and assignment are relatively simple and can be managed by system administrators.

Attribute Constraints: On top of the role foundation, attribute constraints are added to further refine permissions. For example, the administrator role can access all data during working hours but only access part of the data during non-working hours.

3.3.2 Policy Language Design and Application

Policy Language Design: A unified policy language is designed to define the combination rules of roles and attributes. The policy language should be concise and easy to understand, making it easy for system administrators to write and maintain.

Policy Management: Policies are stored in a policy repository, and the system queries and applies relevant policies when a user logs in and accesses resources. The policy repository should have high query performance to ensure the speed of real-time evaluation.

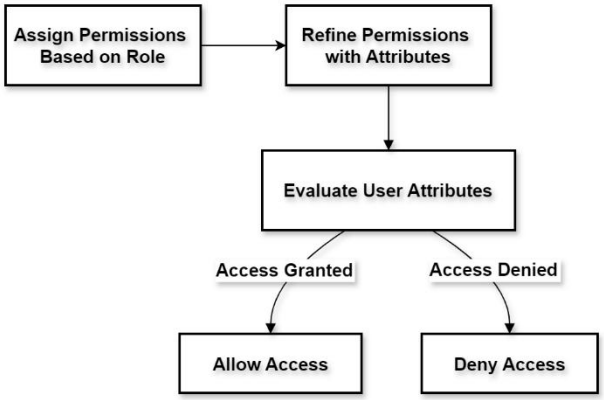

Figure 1. Comprehensive Access Control Flow

Figure 1 outlines the core steps of the comprehensive access control algorithm implementation, focusing on the assignment of permissions based on roles, refinement with attributes, and the final decision to either allow or deny access, culminating in the end of the process.

4. Permission Assignment and Real-time Evaluation

Permission Assignment: When a user logs in, the system assigns permissions based on their role and attributes. First, basic permissions are assigned based on the role, and then refined and adjusted based on attributes. For example, the administrator role can access all data during working hours but only access part of the data during non-working hours.

Real-time Evaluation: When a user accesses resources, the system evaluates their attributes in real-time to determine whether they meet the access policy. If they do, access is allowed; otherwise, access is denied. The evaluation process includes checking the user's, resource's, and environment's current attributes and matching them with the constraints in the policy.

5. Experiment and Analysis

5.1 Experimental Environment and Method

A cloud computing platform is set up in a laboratory environment to simulate a multi-tenant shared resource scenario, and the proposed comprehensive access control algorithm is deployed. The experimental platform includes a user management module, a resource management module, and an access control module.

The performance of the comprehensive access control algorithm is verified through comparative experiments. A control group and an experimental group are set up, with the control group using traditional RBAC or ABAC models and the experimental group using the comprehensive access control algorithm (see figure2).

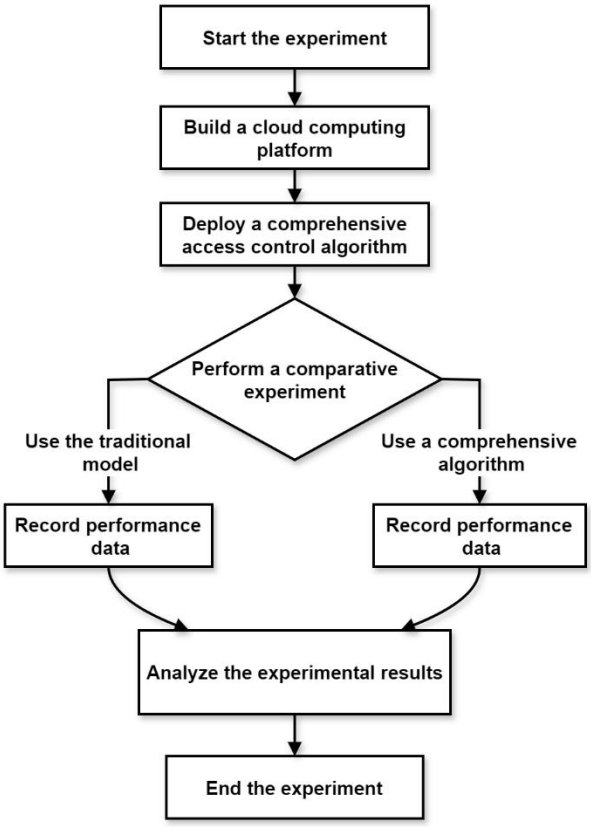

Figure 2. Comprehensive access control algorithm experimental process

5.2 Dataset

A public dataset is selected, containing user information, resource information, and access records. The dataset should be representative and cover different role and attribute combinations. To ensure the comprehensiveness of the experiment, the dataset should include multiple different roles (e.g., administrator, ordinary user, auditor) and multiple different attributes (e.g., access time, access location, resource sensitivity level).

5.3 Comparison Content

Access Control Accuracy: Compare the accuracy of different models in assigning user permissions.

Flexibility: Evaluate the flexibility of different models in responding to dynamic environmental changes.

Response Time: Measure the response time of different models in real-time permission evaluation.

System Overhead: Evaluate the differences in resource consumption between different models.

5.4 Experimental Results

5.4.1 Access Control Accuracy

The comparison results show that the comprehensive access control algorithm significantly outperforms traditional RBAC and ABAC models in terms of permission assignment accuracy (Table I). The comprehensive access control algorithm can perform fine-grained permission assignment based on the combination of roles and attributes, ensuring that users can only access authorized data.

TABLE I. Permission Assignment Accuracy Statistics

Model	Permission Assignment Accuracy
RBAC	85%
ABAC	90%
Comprehensive Access Control	98%

5.4.2 Flexibility

The experiment demonstrates that the comprehensive access control algorithm has higher flexibility in responding to dynamic environmental changes (Table II). The algorithm can adjust user permissions in real-time and dynamically allocate permissions based on changes in environmental attributes.

TABLE II. Flexibility Comprehensive Score Statistics

Model	Flexibility Score
RBAC	60%
ABAC	80%
Comprehensive Access Control	95%

5.4.3 Response Time

The experiment measures the response time of different models in real-time permission evaluation (Table III). The results show that the comprehensive access control algorithm can provide faster response times while maintaining high accuracy and flexibility.

TABLE III. Response Time Comprehensive Statistics

Model	Response Time (ms)
RBAC	20

ABAC	25
Comprehensive Access Control	22

5.4.4 System Overhead

The experiment evaluates the differences in resource consumption among different models (Table IV). The comprehensive access control algorithm has a slightly higher system overhead than RBAC but lower than ABAC, achieving a balance between performance and resource consumption.

TABLE IV. System Overhead Comprehensive Statistics

Model	System Overhead (CPU Usage Rate)
RBAC	10%
ABAC	15%
Comprehensive Access Control	12%

5.5 Experimental Results

Through a detailed analysis of the experimental data, the effectiveness of the comprehensive access control algorithm is verified. The experimental results show that the comprehensive access control algorithm has significant advantages in terms of data security, access efficiency, and system resource utilization. Compared to traditional RBAC and ABAC models, the comprehensive access control algorithm can more flexibly manage permissions, provide finer-grained control, and maintain high response speeds during real-time evaluation [12].

Accuracy: The comprehensive access control algorithm has significantly higher accuracy in permission assignment than RBAC and ABAC models, ensuring data security.

Flexibility: The algorithm has higher flexibility in responding to dynamic environmental changes, allowing for dynamic adjustment of permissions based on changes in environmental attributes.

Response Time: The comprehensive access control algorithm can provide faster response times during real-time permission evaluation, ensuring efficient system operation.

System Overhead: In terms of resource consumption, the comprehensive access control algorithm achieves a balance between performance and resource utilization, with a slightly higher system overhead than RBAC but lower than ABAC.

6. Discussion

6.1 Advantages and Limitations of the Algorithm

6.1.1 Advantages:

The integrated access control model combines the strengths of RBAC (Role-Based Access Control) and ABAC (Attribute-Based Access Control), allowing for dynamic adjustment of user access rights through real-time evaluation of user, resource, and environmental attributes. This flexibility enables the system to adapt to changing business needs and environmental variations, providing more granular access management [13,14]. By further refining attributes, the integrated access control model can achieve more precise access distribution than pure RBAC or ABAC. For example, changes in attributes such as time of day, location, or device can lead to different access allocations, enhancing the security and controllability of the system. Using a unified policy language to define and manage access rules simplifies access management in complex environments. The policy language is easy to understand and maintain, reducing the management complexity for system administrators.

6.1.2 Limitations:

Real-time evaluation of user permissions consumes additional computational resources, especially in scenarios with a large number of users and complex attribute conditions, which may impact system performance. The integrated access control model requires detailed definition of role and attribute combination rules, and the management and maintenance of the policy library may

become complex, particularly in large-scale enterprise environments. Leveraging multi-core processors and distributed computing technologies can break down access control computation tasks into multiple subtasks, processed in parallel to improve computational efficiency. For example, during user login and access allocation, the evaluation of user attributes can be distributed across different processor cores for parallel computation. Introducing caching mechanisms to store frequently accessed permission decision results reduces the number of repetitive calculations. A caching layer can be set up in the system, allowing permission decision results to be retrieved directly from the cache when the same resource is requested again, improving response speed. Optimizing and simplifying access control policies can reduce the number of complex condition evaluations. By analyzing historical access data, redundant and ineffective policy rules can be identified and eliminated, thereby reducing the complexity of policy evaluation. During system idle times, precomputing common access requests and storing the results in cache can reduce the load of real-time computation. When users access the system, they can directly use the precomputed results, reducing the real-time computational burden. Based on the current system load and user access frequency, the allocation of computational resources and the frequency of policy evaluations can be dynamically adjusted. Under high load, priority can be given to processing critical permission decisions, with low-priority tasks deferred. The dependency on environmental attributes may lead to ambiguity in access allocation in some situations; for instance, if accurate environmental attributes cannot be obtained, the system may be unable to make correct permission judgments.

6.2 Comparison with Other Methods

Compared to traditional RBAC and ABAC models, the comprehensive access control model has significant advantages in terms of flexibility and fine-grained control [14]. The RBAC model assigns permissions based on roles, which is simple to manage but lacks flexibility. The ABAC model assigns permissions based on attributes, which is flexible but complex to manage. The comprehensive access control model combines the advantages of both, achieving more flexible and fine-grained permission management, which is suitable for the dynamic cloud computing environment.

TABLE V. Comprehensive Comparison Results

Model	Flexibility	Fine-grained Control	Management Complexity
RBAC	Low	Low	Low
ABAC	High	High	High
Comprehensive Access Control Model	High	High	Medium

In cloud computing environments, the dynamic nature of resources and the need for multi-tenancy make it difficult for traditional access control models to meet security requirements (Table V). The integrated access control model, by combining RBAC (Role-Based Access Control) and ABAC (Attribute-Based Access Control), achieves more flexible and fine-grained access control, better adapting to the needs of cloud computing environments and enhancing system security and management efficiency. The real-time evaluation mechanism of user permissions ensures that in a constantly changing environment, users can only access the resources they are authorized for, preventing unauthorized access.

Multi-Factor Authentication (MFA): When users log in and access critical resources, in addition to the traditional username and password, additional authentication factors such as SMS verification codes, hardware tokens, or biometric identifiers (e.g., fingerprints or facial recognition) can be added. This significantly enhances system security, preventing unauthorized access.

7. Conclusion

This paper proposes an integrated access control model based on the combination of roles and attributes, aiming to address data security issues in cloud computing environments. By integrating the advantages of RBAC (Role-Based Access Control) and ABAC (Attribute-Based Access

Control), the integrated access control model achieves flexible and fine-grained access management. Experimental results demonstrate that this model has significant advantages in terms of data security, access efficiency, and system resource utilization.

Future research can further optimize and expand the integrated access control model in the following areas:

Performance Optimization: To address large-scale users and complex attribute conditions, further optimize the algorithm to reduce the computational overhead of real-time evaluations and improve system performance. Utilize multi-core processors and distributed computing to decompose access control tasks into multiple subtasks for parallel processing. Introduce caching mechanisms to store frequently accessed permission decision results, reducing the number of repeated calculations. During system idle times, precompute common access requests and store the results in cache, allowing users to directly use the precomputed results when accessing the system.

Policy Management Tools: Develop intelligent policy management tools to assist system administrators in more efficiently writing and maintaining policies, thereby reducing management complexity. Design a user-friendly graphical policy editing interface to simplify the process of defining and modifying policies. Introduce automatic policy verification functions to detect conflicts and redundancies in policies in real-time, ensuring the correctness and consistency of policies. Establish a library of common policy templates, enabling administrators to quickly apply and customize policies according to actual needs.

Environmental Awareness Capabilities: Enhance the system's ability to perceive environmental attributes, ensuring accurate acquisition of attribute information in various complex environments, thereby improving the accuracy of access allocation. Integrate various sensor devices to monitor environmental changes in real-time, such as location, time, and device status. Design dynamic update mechanisms to automatically update user attributes and permission information based on environmental changes. Utilize big data analytics to process and analyze massive amounts of environmental data, extracting valuable information for permission decisions.

Security Enhancement: Explore how to combine other security mechanisms (such as multi-factor authentication, behavioral analysis, etc.) to further enhance the system's security.

Validation in Practical Application Scenarios: Validate the effectiveness of the integrated access control model in more practical application scenarios, exploring its potential application in different industries and business contexts.

By exploring these research directions, the integrated access control model is expected to provide more efficient and secure access management solutions in cloud computing environments, safeguarding enterprise data security.

References

[1] L. Sun et al., "BPDAC: A Blockchain Based and Provenance Enabled Dynamic Access Control Scheme," IEEE Access, vol. 11, pp. 142552-142568, 2023, doi: 10.1109/ACCESS.2023.3340887.

[2] K. K. Singamaneni et al., "A Novel Quantum Hash-Based Attribute-Based Encryption Approach for Secure Data Integrity and Access Control in Mobile Edge Computing-Enabled Customer Behavior Analysis," IEEE Access, vol. 12, pp. 37378-37397, 2024, doi: 10.1109/ACCESS.2024.3373648.

[3] M. Tuler De Oliveira et al., "SmartAccess: Attribute-Based Access Control System for Medical Records Based on Smart Contracts," IEEE Access, vol. 10, pp. 117836-117854, 2022, doi: 10.1109/ACCESS.2022.3217201.

[4] Y. Koyasako et al., "Demonstration of Real-Time Motion Control Method for Access Edge Computing in PONs," in Proc. 4th Int. Conf. Front. Technol. Inform. Comput. (ICFTIC), Qingdao, China, 2022, pp. 1020-1023, doi: 10.1109/ICFTIC57696.2022.10075243.

[5] X. Zhang, "Dynamic access control model based on user access behavior in the Internet of Things environment," in Proc. 4th Int. Conf. Front. Technol. Inform. Comput. (ICFTIC), Qingdao, China, 2022, pp. 1020-1023, doi: 10.1109/ICFTIC57696.2022.10075243.

[6] Yang Zongyue. "Data Extraction and Analysis in Intelligent Network Security Attack Detection." Computer Measurement & Control, 2021(05).

[7] Ouyang Yong. "Analysis of Network Security Risks in Broadcasting and Television in 5G Network Environments." China Science and Technology Information, 2021(05).

[8] H. Wang et al., "A distributed ABAC access control scheme based on blockchain," in Proc. 2nd Int. Conf. Comput. Sci. Blockchain (CCSB), Wuhan, China, 2022, pp. 19-25, doi: 10.1109/CCSB58128.2022.00011.

[9] M. Usman et al., "A Blockchain Based Scalable Domain Access Control Framework for Industrial Internet of Things," IEEE Access, vol. 12, pp. 56554-56570, 2024, doi: 10.1109/ACCESS.2024.3390842.

[10] J. Yao et al., "RAN Slice Access Control Scheme Based on QoS and Maximum Network Utility," in Proc. IEEE 6th Adv. Inf. Technol. Electron. Autom. Control Conf. (IAEAC), Beijing, China, 2022, pp. 1853-1858, doi: 10.1109/IAEAC54830.2022.9929619.

[11] F. Liu et al., "Enabling Borderless Office Security: A Comprehensive Zero-Trust Digital Grid Architecture for Flexible Resource Access and Data Protection," in Proc. 9th Annu. Int. Conf. Netw. Inf. Syst. Comput. (ICNISC), Wuhan, China, 2023, pp. 201-204, doi: 10.1109/ICNISC60562.2023.00092.

[12] C.-Q. Kang et al., "Dynamic Access Control Architecture of Distribution Master Station Based on Extended Trust Evaluation," in Proc. IEEE 5th Int. Electr. Energy Conf. (CIEEC), Nangjing, China, 2022, pp. 506-510, doi: 10.1109/CIEEC54735.2022.9846041.

[13] Chen De. "Analysis of Computer Network Security in Cloud Computing Environments." Journal of Jiamusi Vocational Institute, 2021(03).

[14] Wang Peng, Hu Hongbin, Li Yong. "Intelligent Network Security Detection Methods for Big Data Integration Models." Computer Measurement & Control, 2021(05).