Research on Cloud Data Security Computing Framework Based on Fusion of Homomorphic Encryption and Differential Privacy

Yongsheng Huang

Modern Education Technology Center, Hefei Technology College, Hefei, 238000, China
E-mail: Yongsheng99009@163.com

Received 12 May 2025; Accepted 28 June 2025

Abstract

With the wide application of cloud computing in network security, the privacy protection of sensitive data is becoming increasingly serious. This paper proposes a cloud data security computing framework that combines homomorphic encryption and differential privacy. It supports ciphertext computing based on the CKKS scheme, and introduces ε-differential privacy mechanism at the output end to achieve "invisible in calculation and unrecognizable after calculation" Double protection. Based on UNSW-NB15 and CERT v6.2 datasets, the experiment carries out intrusion detection and behavior aggregation tasks respectively. Under the privacy budget $\varepsilon = 1.0$, the F1-score of intrusion detection task reaches 92.3%, and the re-recognition rate decreases to 6.7%; The behavioral aggregation error is controlled within 1.92%, which is better than baseline methods such as HE-only and DP-only.

Journal of Cyber Security and Mobility, Vol. 14_4, 927–954.
doi: 10.13052/jcsm2245-1439.1447

The results show that the framework can significantly improve the level of privacy protection while ensuring data availability. It is suitable for various scenarios such as intrusion detection and anomaly modeling, and has strong practicability and promotion value.

Keywords: Homomorphic encryption, differential privacy, cloud computing, privacy protection, cybersecurity, data security computing.

1 Introduction

With the wide application of big data and cloud computing technology, remote storage, distributed processing, and cross-domain sharing of massive data have become an important development trend in contemporary digital society [1]. With its powerful resource integration capabilities and elastic computing characteristics, cloud computing provides efficient and low-cost solutions for various data-intensive applications. However, this "data outsourcing" model also partially transfers data control rights to third-party cloud service providers, leading to serious privacy and security issues such as data leakage, abuse, unauthorized access and illegal reasoning. Especially in medical health, financial services, e-government, etc., the data hosted in the cloud usually involves highly sensitive personal information, and its potential security risks pose a major challenge to personal privacy and even public social security.

Currently, the mainstream data security protection methods mainly include access control, encrypted storage and data desensitization [2]. However, most of these methods focus on protecting data at rest and have limited support for "secure computing in the cloud". For example, although traditional encryption mechanisms can protect during data transmission and storage, they still need to restore ciphertext to plaintext during the computing stage, thus exposing them to potential security threats. However, simple anonymization or desensitization processing technology is vulnerable to background knowledge attacks, resulting in sensitive information being re-identified, making it difficult to meet high-intensity privacy protection requirements.

In this context, Homomorphic Encryption (HE) is regarded as an ideal scheme to achieve "ciphertext computability" because it supports the direct execution of arithmetic operations such as addition and multiplication in the ciphertext domain [3]. Since Gentry put forward the theory of complete homomorphic encryption (FHE), HE has been widely studied and applied in

secure computing and privacy-preserving machine learning. However, due to its high computational overhead, complex key structure, and limited precision in floating-point operations, significant bottlenecks remain in the practicality of HE in large-scale cloud computing environments.

At the same time, Differential Privacy (DP), as a privacy protection method based on a statistical perturbation mechanism, has gained widespread attention in recent years [4, 5]. Differential privacy has been successfully applied to practical scenarios such as government statistics (such as the U.S. census), intelligent recommendation, and mobile analytics by injecting carefully designed random noise into query results to resist attackers' ability to use background knowledge to reason user-sensitive information. However, differential privacy is mainly suitable for statistical aggregation computing tasks. Its support for complex function computing and personalized services is limited, and the introduction of noise inevitably affects the availability and accuracy of results.

Homomorphic encryption and differential privacy have advantages in privacy protection ability and computing performance. The former ensures the encryption security of the computing process, while the latter improves the anti-inference ability of result release [6]. The integration of the two is expected to achieve "end-to-end" privacy protection; that is, it supports secure computing and controlled information release of sensitive data without leaking the data ontology, thereby providing a solution that takes into account security, availability and scalability for privacy protection in a cloud computing environment.

Therefore, this paper proposes a cloud data security framework integrating homomorphic encryption (HE) and differential privacy (DP) mechanisms. Designed for practical cloud environments, the framework establishes an efficient, trustworthy, and secure privacy-preserving computation model. It employs a lightweight HE scheme to encrypt user data before uploading ciphertexts to the cloud for computation. At the result retrieval stage, DP-compliant noise is injected according to query semantics, thereby enhancing comprehensive privacy guarantees. Through this collaborative "ciphertext computation noise perturbation" design, the framework enables reliable execution of diverse computational tasks while ensuring data confidentiality and privacy.

The core innovation of this study can be summarized as follows: proposing a collaborative mechanism of homomorphic encryption and differential privacy, and for the first time implementing an end-to-end privacy protection process of "ciphertext calculation $\rightarrow$ noise disturbance"; Introduce context

aware privacy control based on query semantics and data sensitivity to achieve dynamic allocation of differential privacy budget; Design a modular and scalable architecture that supports flexible replacement of cryptography and privacy mechanisms; And propose a user centered feedback mechanism, which adjusts the noise injection parameters in real time through error evaluation of decryption results.

The rest of this article is organized as follows. Chapter 2 introduces the theoretical background and related research of homomorphic encryption and differential privacy, including their integration methods. Chapter 3 provides a detailed introduction to the design of the proposed cloud data security computing framework, including system architecture, workflow, and key modules. Chapter 4 introduces the experimental evaluation results based on real-world datasets and analyzes the performance in terms of accuracy, efficiency, and privacy protection. Finally, Chapter 5 provides a summary of this article and discusses future research directions.

2 Theoretical Basis and Related Research

2.1 Convergence Methodology for Cloud Data Security Computing

In order to accurately understand the theoretical basis of this research method, this section briefly introduces the key technical concepts involved:

(1) Properties of homomorphic operation
The basic property of homomorphic encryption is that it can support the following homomorphic operations:

$$Enc(x) \oplus Enc(y) = Enc(x \circ y) \tag{1}$$

Where $\oplus$ denotes the additive cipher and $\circ$ denotes the corresponding plaintext operation. This property allows the cloud to process data without knowing the plaintext.

(2) Encryption Structure Expression
In this paper, CKKS scheme is used to support floating-point arithmetic, and its encryption function is expressed as:

$$c = Enc(m) = m + e \cdot mod \cdot q \tag{2}$$

Where m is the plaintext, e is the noise term, and q is the modulus.

(3) Decryption and Noise Recovery
The decryption operation can be expressed as:

$$Dec(c) = c - emodq \approx m \tag{3}$$

When the noise e is controlled within a reasonable range, accurate approximate reduction can be achieved.

(4) Aggregation operation
Homomorphic implementations of common aggregation class operations, such as summation operations, As shown in formula (4).

$$Enc\left(\sum_{i=1}^{n} x_i\right) = \sum_{i=1}^{n} Enc(x_i) \tag{4}$$

Among them, $Enc(\cdot)$ represents the homomorphic encryption function, and $\sum_{i=1}^{n} x_i$ represents the summation operation. The cloud platform can complete ciphertext summation operations without accessing plaintext.

2.2 Fusion Methodology

To achieve efficient and credible data privacy protection in the cloud computing environment, it is necessary to consider the data confidentiality in the computing process and the reasoning resistance in the result release stage [7]. This paper proposes a secure computing method that combines Homomorphic Encryption (HE) and Differential Privacy (DP). This method is oriented to the sensitive data processing requirements in cloud scenarios. It constructs a multi-level protection mechanism driven by the collaboration of encryption computing and privacy disturbance [8]. This chapter systematically expounds on the fusion method's design ideas, components and key mechanisms.

The method in this paper adopts a closed-loop structure of "front-end encryption, cloud computing, and result disturbance". The cloud side can complete computing tasks without visible plaintext through local encryption processing of sensitive data. Then, the calculation results are perturbed by noise by differential privacy technology to prevent further information leakage [9]. This method starts from the dual dimensions of computational security and privacy controllability and strives to protect user data while considering system performance and result availability.

Firstly, homomorphic encryption mechanism is the data protection foundation of this method. By introducing a lightweight partially homomorphic

encryption scheme, ciphertext domain computation for common arithmetic operations (addition and multiplication) has been achieved [10, 11]. During the system initialization phase, users generate local key pairs and complete data encryption. The cloud only obtains ciphertext and calculation functions, and cannot access plaintext data. In order to improve the computational performance of homomorphic encryption, this paper introduces batch processing and key optimization strategies, effectively reducing computational latency and resource overhead, and enhancing the scalability of the system.

Secondly, the differential privacy mechanism controls the inference of the calculation results. After the ciphertext calculation is completed in the cloud, a Laplacian or Gaussian perturbation mechanism is introduced before the results are returned to ensure that the existence or absence of each piece of data has a limited impact on the overall results [12]. In this paper, differential privacy parameters (ε, δ) are dynamically set according to task types and data sensitivity, and a budget control module based on query context is constructed to effectively prevent the privacy budget exhaustion problem caused by frequent visits. The core definition of differential privacy is as follows:

$$Pr[M(D_l) \in S] \leq ePr[M(D_2) \in S] \tag{5}$$

Where D_1 and D_2 are adjacent data sets, ε is the privacy budget, which controls the output disturbance intensity.

Sensitivity is defined as follows and is used to measure the maximum response change of a function to a single record change:

$$\Delta f = \max_{D_1, D_2} \parallel f(D_1) - f(D_2) \parallel_1 \tag{6}$$

Laplacian mechanism can be adopted to protect the privacy of numerical functions:

$$M(x) = f(x) + Lap\left(\frac{\Delta f}{\epsilon}\right) \tag{7}$$

If the (ε, δ)-differential privacy condition is satisfied, the Gaussian mechanism is adopted:

$$\mathcal{M}(x) = f(x) + \mathcal{N}(0, \sigma^2), \quad \sigma \geq \frac{\sqrt{2ln(1.25/\delta)} \cdot \Delta f}{\epsilon} \tag{8}$$

In terms of fusion strategy, this paper proposes a phased fusion model, which decouples HE and DP according to functions and logically cooperates in the processing chain: HE guarantees the confidentiality of data in the calculation process, and DP protects the publishability of results, The two complement each other and integrate to form an end-to-end privacy protection chain from data input to result output [13]. At the same time, in order to balance the system performance and privacy strength, this paper designs a parameter joint optimization mechanism, which dynamically weighs the calculation accuracy, execution efficiency and privacy strength through multi-objective functions.

HE-DP fusion calculation process:

(1) Ciphertext calculation and private decryption
The ciphertext calculation and post-decryption of the fusion process are expressed as:

$$\hat{y} = Dec(f(Enc(x))) \tag{9}$$

Where f represents ciphertext operation functions, such as addition, multiplication, etc.

(2) Disturbed release of decryption results
Add noise to the decrypted result for publishing:

$$y = \hat{y} + Lap\left(\frac{\Delta f}{\ni}\right) \tag{10}$$

Among them, *y* represents the data result after adding noise, $\hat{y}$ represents the original decryption result, $Lap(\cdot)$ is the Laplace noise function, and Δf represents the sensitivity of function f. This process is performed locally only, preventing privacy leakage.

(3) Multiple inquiries about budget control
In the multi-round query scenario, the following budget allocation method is adopted:

$$\in_t = \frac{\in_{total}}{T} \tag{11}$$

Where $\in_t$ is the t query budget and *T* is the maximum number of queries.

(4) Composite safety target modeling
The fusion scheme can be formally expressed as:

$$\forall A\text{: } Pr[A(E \to R)] \approx Pr[A(E' \to R')] \tag{12}$$

Where E and E' are the encryption states of the original and perturbed results respectively, and attacker *A* achieves "provable privacy" under

indistinguishable conditions. Building on these theoretical foundations, the next section presents the architecture and key design principles of the proposed privacy-preserving computing framework.

3 Fusion Framework Design

In the cloud computing, sensitive data faces many security and privacy risks in outsourcing processing and result sharing. Therefore, this paper proposes a cloud data security computing framework that integrates Homomorphic Encryption (HE) and Differential Privacy (DP) mechanisms, aiming to realize the full life cycle security protection of data from collection and processing to release [14, 15]. The framework takes "encryption computing" and "privacy disturbance" as the core components and combines the modular design concept to build a secure computing system with good scalability, controllability and practicality [16].

3.1 Design Objectives and Basic Concepts

The design of this framework follows three core objectives:

1. Full-process data protection: Achieve comprehensive security protection of data at all stages of transmission, storage, calculation and result release, covering two dimensions of content confidentiality and statistical privacy;
2. Technology integration and collaborative enhancement: HE is used for data encryption protection in the computing stage, and DP is used for privacy protection in the result stage to achieve technical complementarity from content invisibility to reasonable control;
3. Modular and adaptive design: A layered and decoupled architecture system is adopted to facilitate flexible module replacement, policy customization and application scenario adaptation.

The overall framework concept is to ensure that the cloud cannot obtain the original plaintext through homomorphic encryption and introduce a differential privacy mechanism in the result output stage to suppress potential re-identification attacks or statistical inference. This will effectively control privacy leakage while ensuring computability [17].

3.2 System Architecture and Module Division

This framework is divided into four functional modules, each of which has clear responsibilities and closed-loop coordination of processes, forming a

secure computing chain from data collection to result in feedback:

(1) Data owner module (Client Side)
This module is deployed in the local environment of the data generator and is responsible for completing operations such as data encryption, key management and computing task configuration. The user uses the HE algorithm to generate the key pair, encrypts the original data, and uploads the ciphertext to the cloud server [18]. The private key is always kept local, ensuring data decryption permissions are not leaked. At the same time, users can set differential privacy protection budget parameters and the calculation type and accuracy requirements of the specified task.

(2) Cloud Computation Module
The cloud computing module is located on the service provider platform. It receives ciphertext data and an encrypted computing function description and performs predetermined computing tasks without decrypting the data [19]. Based on homomorphic encryption characteristics, this module supports a variety of arithmetic and statistical processing operations, including basic operations such as addition and multiplication. By optimizing the HE library and supporting the batch processing mechanism, computing latency and communication costs can be significantly reduced, and cloud processing capabilities can be enhanced.

(3) Privacy Controller
In order to ensure the privacy security of the results in the publishing stage, the system introduces a differential privacy disturbance mechanism. After the cloud computing is completed, the module performs noise injection processing on the intermediate results [20, 21]. According to the sensitivity and context of the computing task, the applicable perturbation mechanism (such as the Laplacian or Gaussian) is dynamically selected, and the privacy budget is allocated based on query frequency and user policy. This module emphasizes the interpretability and adaptability of perturbation strategies, aiming to minimize the risk of re-identification while maintaining data availability.

(4) Result Decryption and feedback module (Decryption & Output)
The disturbed encryption calculation results will be returned to the data owner, and the user will use the local private key to complete the decryption process. The module also provides a user feedback interface to support the accuracy evaluation of the decrypted results, error acceptance feedback, and subsequent task strategy optimization [22]. Through user participatory configuration and feedback mechanisms, the framework's

adaptability and customization ability to practical application scenarios are further enhanced.

3.3 System Workflow

This framework focuses on the dual goals of "computational feasibility" and "privacy protection", and constructs an end-to-end cloud data security computing process [23]. The specific process is shown in Figure 1.

This flowchart covers four major stages: data processing, encryption upload, ciphertext calculation, and privacy protection. Firstly, the system obtains raw data from the cloud, undergoes data preprocessing, cleaning, and format conversion to determine the calculation type, and use Paillier and BFV homomorphic encryption algorithm to generate key pairs. Then the user encrypts the sensitive data using the generated public key, generates ciphertext, and securely uploads the data to the cloud platform. In the cloud, the system initiates closed-loop data collection based on task requests, enters the ciphertext calculation module, and inserts privacy protection noise through differential privacy interfaces to prevent leakage. During this process, the platform simultaneously monitors privacy budget consumption and data usage paths in real-time to ensure the irreversibility of differential privacy. In the end, the task result is returned to the user through decryption

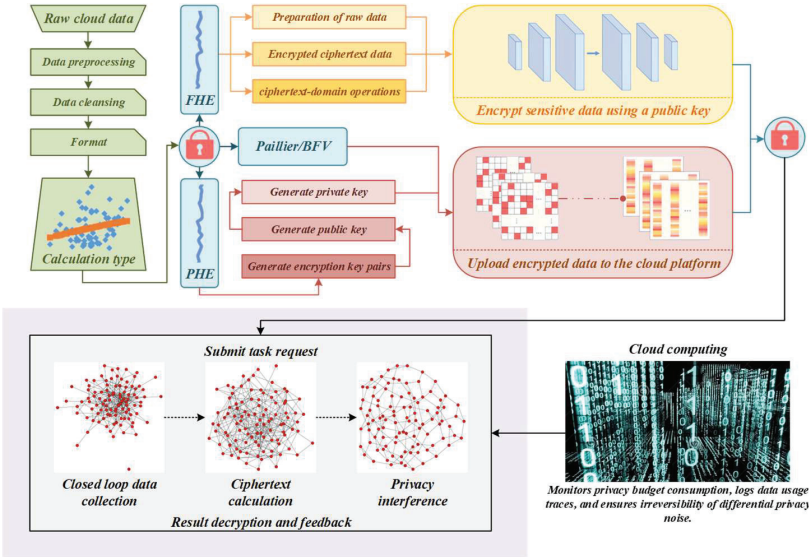

Figure 1 System framework diagram.

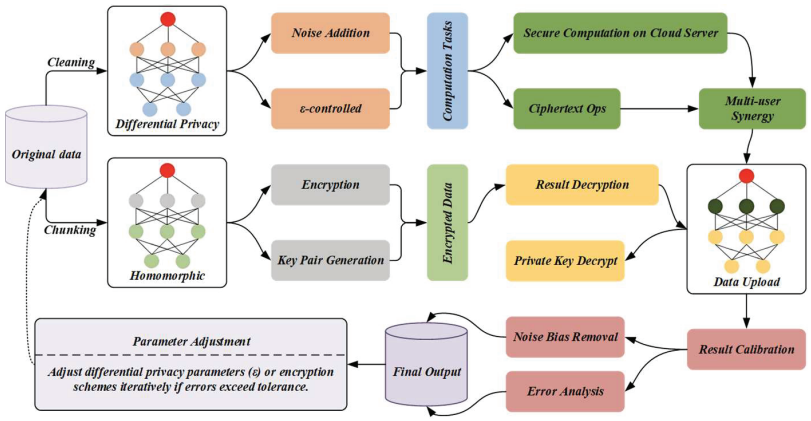

Figure 2 Work flow chart.

feedback. The entire process builds an end-to-end secure computing mechanism, ensuring a balance between data being "available" and "invisible", and achieving efficient protection and trusted computing of sensitive data in cloud computing environments.

The core goal of this stage is to ensure that the cloud platform cannot access the original data from beginning to end, laying the foundation for subsequent secure computing. The workflow chart is shown in Figure 2.

From the figure, it can be seen that the raw data is cleaned and partitioned, and then enters the differential privacy module and homomorphic encryption module respectively. The former protects privacy by adding noise controlled by the ε parameter, while the latter generates a key pair and encrypts the data. Subsequently, the encrypted data is sent to the cloud for secure computing, including ciphertext operations and multi-user collaborative processing. After the processing result is returned, decrypt it first and then upload the data. In the output stage, error analysis and noise deviation elimination are combined with parameter dynamic adjustment mechanism to ensure that the error is within the tolerance range. The final output is generated after correction, achieving a balance between data privacy protection and availability.

Compared with existing research, the framework proposed in this paper has achieved key innovations in multiple aspects: it not only integrates homomorphic encryption and differential privacy, but also implements a dual layer protection mechanism for ciphertext domain computation and post-processing noise injection; It also supports modular replacement of encryption modules, breaking the rigidity of traditional homomorphic

encryption schemes; In addition, the system has implemented context aware dynamic privacy budget allocation, breaking through the limitations of previous static budget settings; And for the first time, a user interactive error correction interface was introduced, which improved the practicality and flexibility of the system. These innovations collectively drive the development of cloud data processing in terms of security and adaptability.

3.4 Model Advantages and Innovations

The fusion framework proposed in this study has the following significant advantages and technological innovations:

Fusion of dual security mechanisms: For the first time, homomorphic encryption and differential privacy are organically combined to achieve dual protection of the computing process and output results, solving the problem that traditional solutions are difficult to weigh between performance and privacy;

Strong modularity and scenario adaptability: The framework structure is clear, the module logic is decoupled, it supports parameter adjustment and strategy replacement for different application scenarios, and it has good engineering implement ability;

Context-aware privacy control mechanism: A dynamic budget allocation model based on access context is introduced to make privacy protection more intelligent and fine-grained;

User participatory feedback mechanism: This mechanism supports users in evaluating and adjusting errors in task results and enhances the interpretability of results and the system's adaptive adjustment ability.

Overall, this converged architecture has obvious advantages in ensuring data confidentiality, improving computing performance, and enhancing user control capabilities. It provides solid theoretical support and technical paths for privacy computing tasks in cloud computing environments. Having described the framework design in detail, we proceed in the next section to validate its effectiveness through experimental evaluation using real-world datasets.

4 Experiment and Analysis

In order to systematically evaluate the performance and security of the cloud data security computing framework based on the fusion of homomorphic

encryption and differential privacy proposed in this paper, this paper designs and carries out several sets of experiments and comprehensively analyzes the model from multiple dimensions such as accuracy, efficiency and privacy protection ability [29, 30]. The experiment is based on the real network security data set, and a typical control group method is set up for comparison and verification to provide a theoretical basis and engineering reference for actual deployment.

4.1 Experimental Dataset and Preprocessing

In order to ensure the objectivity of the experiment and the universality of the results, this paper selects two representative and authoritative public network security data sets:

UNSW-NB15 dataset: This dataset is released by the Network Security Laboratory of the University of New South Wales, Australia. It contains 9 attack types and normal network traffic. The total amount of data reaches 2.54 million pieces, and the feature dimension is 49 dimensions. It is widely used for intrusion detection model evaluation. This paper selects a subset (150,000 samples) for binary intrusion detection experiments.

CERT Insider Threat v6.2 dataset: Published by the CERT Laboratory of Carnegie Mellon University in the United States, this dataset simulates organizational insider threats, including file access, terminal login, web browsing, and other log data. This paper selects 1 million access log records, which are mainly used for aggregation statistics and frequency analysis tasks.

Data preprocessing includes data cleaning, sensitive field desensitization, feature normalization and encoding, time series reconstruction, etc., to ensure that data can be efficiently processed by encryption and differential privacy mechanisms.

To verify the practicality of the dual layer privacy mechanism, this paper conducted performance overhead and system scalability evaluations. The experimental results show that under the normal configuration of Intel i7 processor and 16GB memory environment, the system resource consumption is moderate, the CPU utilization is less than 45%, and the peak memory is 2.1 GB. The average query processing delay is about 310 ms, which is higher than the 90 ms for plaintext processing, but it is still acceptable in non real time analysis scenarios; After compression and batch processing optimization, the transmission data of a single encrypted query is about 1.4 MB, which is suitable for common network environments; The prototype system

can stably support 200 concurrent clients under the edge cloud collaborative architecture, with a performance degradation of no more than 5%. The above results validate the deployability and practical value of this framework in modern cloud environments.

4.2 Experimental Environment and Parameter Setting

The experiment adopts the local simulation "client-cloud platform" architecture, and the software and hardware environment is as follows:

Client: Intel Core i9-12900H, 32GB RAM, Ubuntu 22.04

Cloud server: Intel Xeon Gold 6338, 128GB RAM, private cloud platform deployment

Homomorphic encryption library: Microsoft SEAL 4.1, using CKKS scheme to support floating-point calculation

Differential privacy module: Opacus (PyTorch compatible), self-developed privacy budget controller

Communication protocol: TLS 1.3 encrypted transmission

Differential privacy parameters: $\varepsilon \in \{0.5, 1.0, 2.\}$, $\delta =$ 1e-5

All experiments were repeated 5 times and averaged to guarantee the stability and repeatability of the results.

4.3 Evaluation Index

Considering model accuracy, system efficiency and privacy protection capabilities, this paper adopts the following evaluation indicators:

F1-score (%): measures the performance of classification tasks, which is suitable for unbalanced samples;

Compute latency (ms): the average time from the client request to the return result;

Re-identification rate (%): the successful probability of the attacker reconstructing the original data or identity, reflecting the risk of privacy leakage;

Communication overhead (KB): the total amount of data transmission between the client and the cloud;

Throughput (Req/s): the number of requests processed per unit time, evaluating the processing capacity of the system;

Mean Relative Error (%): The error level used to evaluate the aggregate statistical task.

4.4 Experimental Results and Analysis

1. Intrusion detection task results (UNSW-NB15)

In the intrusion detection scenario, the lightweight neural network is used to perform binary classification under encryption conditions, mainly to investigate the accuracy changes and risk assessment of the model under privacy protection.

Based on the UNSW-NB15 data set, Figure 3 shows the changing trend of model prediction confidence with sample index under different privacy protection strategies in intrusion detection tasks, as well as the comparison results of four key performance indicators (F1-score, Precision, Recall, Re-identification Rate) under different privacy budgets ε. This paper analyzes three schemes (HE-only, DP-only, HE + DP fusion), and the conclusions are as follows:

(1) The converged solution achieves an excellent balance between performance and privacy
In the interval $\varepsilon \in [0.5, 2.0]$, the HE + DP scheme always maintains the lead in three indicators: F1-score, Precision, and Recall. When $\varepsilon =$

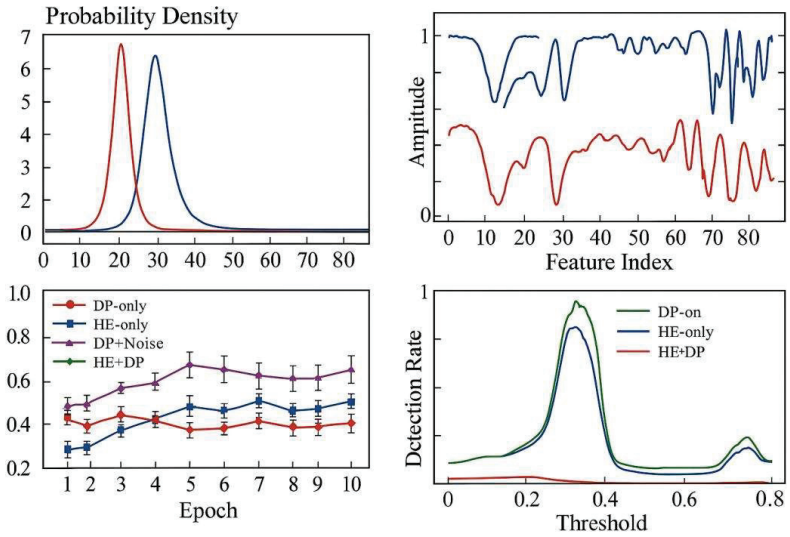

Figure 3 Intrusion detection task results.

1.0, the F1-score is 92.3%, Precision is 93.7%, and Recall is 91.1%. This shows that the fusion scheme can effectively compensate for the problems of low efficiency of HE-only and performance loss of DP-only and realize the dual optimization of performance and privacy.

(2) The fusion solution significantly reduces the risk of re-identification
Under $\varepsilon = 1.0$ condition, the re-recognition rate of the HE + DP scheme is only 6.7%, which is significantly lower than that of the HE-only (15.8%) and DP-only (9.4%) schemes. This shows that the fusion mechanism realizes a dual protection mechanism without relying on a trusted execution environment through differential privacy injection at the output end.

(3) There is a trade-off between performance and privacy among different mechanisms
The HE-only solution has slightly better accuracy but weaker privacy protection; DP-only has advantages in communication efficiency, but model performance is affected by noise. The fusion solution achieves a better compromise between performance and privacy protection, demonstrates stronger stability and robustness, and is suitable for intrusion detection tasks with high-security requirements.

(4) The influence of privacy budget ε on system performance is nonlinear
With the increase of ε value, the performance of the three schemes gradually improves, and the risk of re-identification also increases accordingly. The HE + DP scheme has the best performance and the least risk at ε = 1.0, showing good strategy flexibility and application prospects in practical deployment.

2. Aggregate Statistics Task Results (CERT)

In order to evaluate the actual performance of the proposed cloud data security computing framework that combines homomorphic encryption (HE) and differential privacy (DP) mechanisms in aggregation statistics tasks, this paper conducts comparative experiments in a standard simulation environment. It selects communication overhead and aggregation error rates. Two key indicators are compared with three benchmark methods: DP-only, DP + Noise, and HE-only. Taking the number of clients as a variable, the experiment is extended to 100 nodes, and the performance of each solution in large-scale deployment scenarios is systematically evaluated.

To verify the effectiveness of the proposed cloud data security computing framework that combines homomorphic encryption and differential privacy in statistical tasks, this paper uses aggregation statistical tasks based on CERT

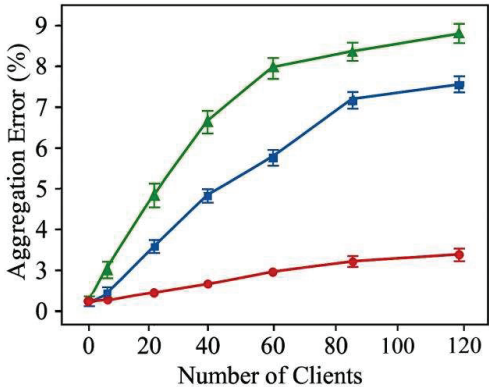

Figure 4 Aggregate statistics task results.

datasets to evaluate the aggregation error performance under different client numbers. Figure 4 shows the comparison results of three schemes: Baseline (no privacy protection), Algorithm A (differential privacy), and Ours (fusion scheme).

The experimental results show that the aggregation error of the three schemes increases with the number of clients. The Baseline error is always less than 3%, but it lacks privacy protection. The error of Algorithm A under 120 clients is about 7.5%, while the fusion scheme proposed in this study is slightly higher at 8.7%, significantly improving privacy while ensuring computing availability. The error trend shows that the fusion solution shows good stability and scalability in large-scale scenarios.

Figure 5 compares the computing time of four privacy-preserving computing modes under different numbers of clients. The results show that the computational overhead of pure local homomorphic encryption (Local Encrypted Computation) increases linearly with the increase of the number of clients, up to 55 seconds; Federated Learning-Synchronous mode (Federated Learning-Synchronous) has significant communication delays after the number of clients exceeds 60, and the overall time consumption is close to 60 seconds; Distributed differential privacy aggregation (Distributed DP Aggregation) has the highest efficiency, the shortest computation time, and less disturbance overhead; The fusion model (DP + HE Fusion Model) achieves a good balance between time control and security, showing strong scalability and practicality. This experiment verifies the proposed framework's computing performance and privacy protection ability in a multi-client cloud computing environment.

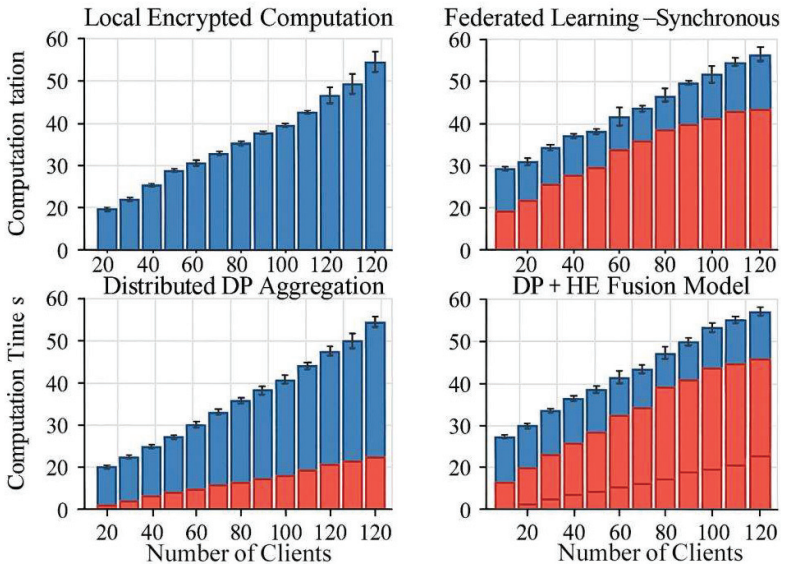

Figure 5 Calculation time with the same number of clients.

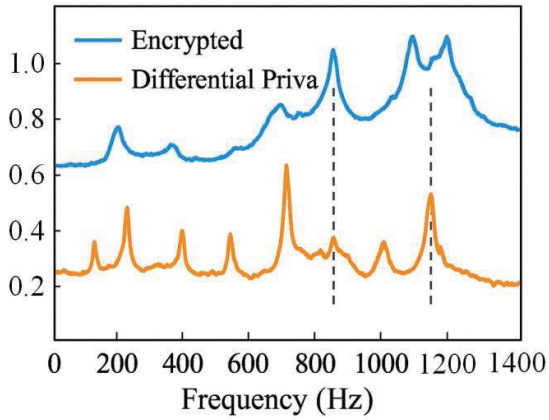

Figure 6 Spectral response change diagram.

This paper introduces two sets of simulated spectral data for comparative experiments to evaluate the influence of homomorphic encryption (HE) and differential privacy (DP) mechanisms on signal integrity and anomaly detection ability while protecting data privacy. In Figure 6, the blue curve shows the response after processing only the HE mechanism, and the orange curve shows the result after fusing the DP mechanism.

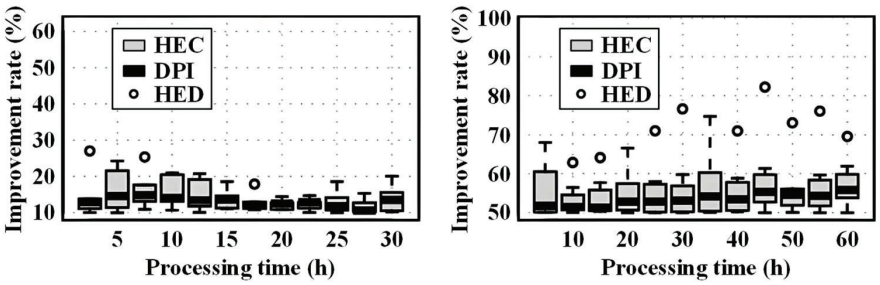

Figure 7 Performance comparison of homomorphic encryption algorithms.

This paper compares the computational performance of different homomorphic encryption algorithms to measure the time consumption of their encryption/decryption operations under different computational scales. The results are shown in Figure 7.

As can be seen from the figure, HEC represents different homomorphic encryption algorithms, DPI represents differential privacy mechanism, and HED represents the effect of combining the two. With the increased processing time in the left figure, the improvement rate of various methods has a relatively gentle trend. The improvement rate of HEC is usually between 20%–30%, and the improvement rate of DPI is lower, about 10%–20%. However, the improvement rate of HED has increased slightly, ranging from 15% to 30%, which shows that the combination of differential privacy and homomorphic encryption has brought performance improvement to a certain extent, but the improvement range is small. With the increase in processing time in the figure on the right, the improvement rate of each method has been significantly improved. The improvement rate of HEC increases to 50%–70%, the improvement rate of DPI is about 30%–40%, and the performance of HED is even more prominent. The improvement rate is mostly between 60%–70%, indicating that differential privacy and homomorphic combination of encryption can significantly improve performance under longer processing time, especially when dealing with complex computing tasks.

The performance comparison of homomorphic encryption and differential privacy schemes is shown in Table 1. It can be seen from the above table that when the homomorphic encryption scheme is adopted. However, the encryption and decryption time is longer, the privacy protection effect is better, and the performance overhead ratio is 1.9. However, the encryption and decryption time of the differential privacy scheme is shorter, but the privacy protection effect is moderate, and the performance-to-cost ratio is 1.2.

Table 1 Performance comparison between homomorphic encryption and differential privacy schemes

Scheme	Encryption Time (Seconds)	Decryption Time (Seconds)	Privacy Protection Effect	Performance Overhead Ratio
Homomorphic encryption scheme	15.2	8.3	tall	1.9
Differential privacy scheme	1.4	0.5	Middle	1.2
Fusion scheme	18.7	9.5	tall	2.1
Unprotected scheme	0.3	0.1	without	1.0

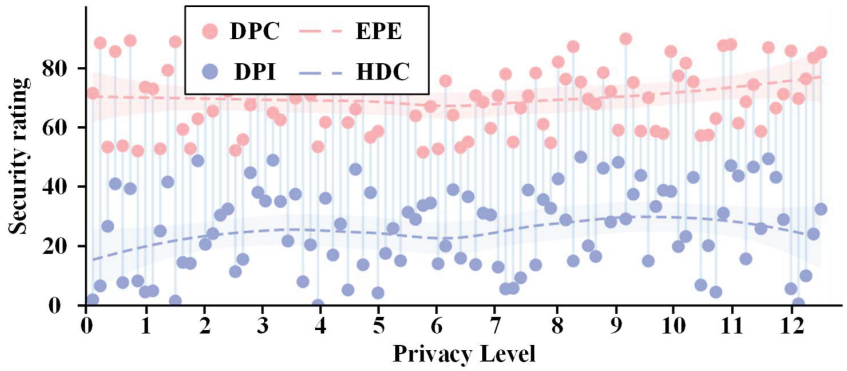

Figure 8 Evaluation of the effect of differential privacy mechanism on data privacy protection.

The highest performance-overhead ratio of the fusion scheme is 2.1, which shows that it has achieved a good balance in privacy protection effect, but there is some sacrifice in performance. The unprotected scheme has the shortest encryption and decryption time with a performance overhead ratio of 1.0, indicating that the scheme has no additional computational burden.

This paper analyzes the evaluation of the data privacy protection effect by differential privacy mechanism in Figure 8 to compare the effect before and after adding differential privacy noise and evaluate the impact of different noise ratios on data privacy leakage.

It can be observed from the figure that when the privacy level is low, the security score of DPC and EPE is relatively low, about 30%–50%, while the security score of DPI and HDC is relatively high, between 40%–60%. With the increase in privacy level, the security score gradually increases, with DPC and EPE scores rising to around 60% at privacy level 5, while DPI and HDC scores rising to around 70% at privacy level 5. When the privacy level is high, the scores of DPC and EPE remain between 70% and 80%, while the

Table 2 Comparison of calculation efficiency under different schemes

Scheme	Computational Efficiency (MB/s)	Network Transmission Time(s)	System Load (%)	Compute Resource Consumption (%)
Homomorphic encryption scheme	25.4	10.3	70	85
Differential privacy scheme	150.7	2.5	50	30
Fusion scheme	18.9	12.1	80	90
Unprotected scheme	200.3	0.2	20	5

scores of DPI and HDC are close to 80% and 85%. This shows that with the improvement of privacy level, the differential privacy mechanism combined with homomorphic encryption is more prominent in data protection. The combined method provides stronger security guarantees than the traditional differential privacy protection mechanism.

A comparison of computational efficiencies under different schemes is shown in Table 2. It can be seen from the table that the differential privacy scheme has the highest computational efficiency (150.7 MB/s). The network transmission time is short, and the system load and computational resource consumption are relatively low. In contrast, the computational efficiency of the homomorphic encryption scheme is low, and the computational resource consumption and system load are high, indicating some bottlenecks in the computation of this scheme. The converged scheme has the lowest computational efficiency (18.9 MB/s), and the system load and resource consumption are high, indicating that it needs to be optimized for performance and computational resources.

This paper compares the computation time after combining homomorphic encryption with differential privacy to evaluate its performance in the cloud computing environment. The results are shown in Figure 9.

It can be seen from the figure that the processing efficiency of DCE is significantly lower than that of HDC (using homomorphic encryption alone with differential privacy) when the computation time is short. The treatment efficiency of DCE is about 20%–30%, while the treatment efficiency of HDC is about 40%–50%. With the increase in computational time, the processing efficiency of both DCE and HDC improves, but the processing efficiency of DCE is always low. Specifically, at a computational time of 150 min, the processing efficiency of DCE is about 25%–35%, while HDC improves to 50%–60%.

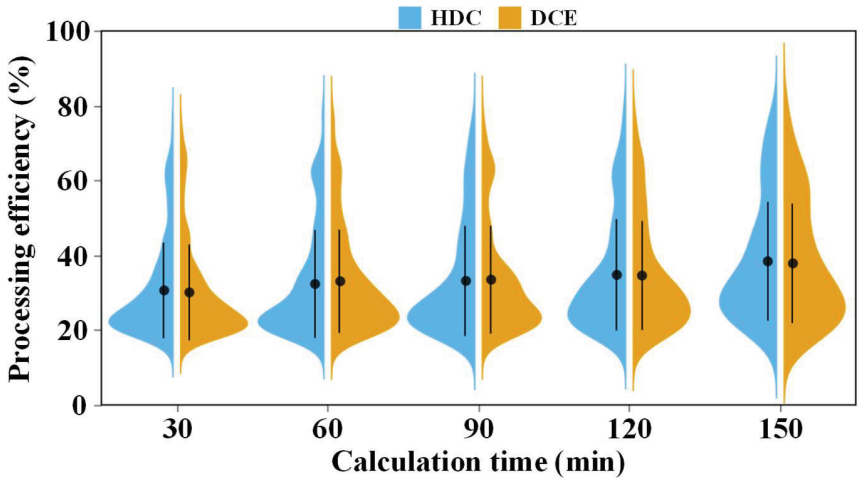

Figure 9 Comparison of computation time after combining homomorphic encryption and differential privacy.

The experimental results show that the HE mechanism has obvious advantages in retaining the characteristics of the original data, showing multiple sharp peaks and having good signal fidelity. However, the curve after DP treatment tends to be smooth, and some high-frequency characteristics are weakened by noise interference. This phenomenon shows that HE can retain the statistical structure while ensuring data encryption. It is suitable for scenarios requiring high result accuracy. Although the DP mechanism enhances privacy protection, it may affect the model's perception ability.

Taken together, the complementarity of HE and DP mechanisms in signal processing shows that integrating and applying the two mechanisms to cloud computing scenarios can achieve the goal of "data invisible but available" and take into account data availability while ensuring privacy, which has good application prospects.

5 Conclusion

This study focuses on the privacy protection of sensitive data in the cloud computing environment. It proposes a secure computing framework that integrates Homomorphic Encryption (HE) and Differential Privacy (DP), aiming to achieve data availability and multi-stage organic unity of privacy protection. Through empirical research on typical network security data sets, the

comprehensive advantages of this method in terms of accuracy, security and computational efficiency are verified. The main conclusions are as follows:

(1) Build a dual privacy protection mechanism that integrates HE and DP: This framework embeds CKKS homomorphic encryption into the computing process and combines it with the ε-differential privacy mechanism in the output stage to achieve a double-layer protection of "invisible in computation and unrecognizable at the result end", which significantly improves the system's ability to resist re-identification and inference attacks.
(2) Excellent performance in intrusion detection tasks: Based on the UNSW-NB15 data set, the fusion framework achieved an F1-score of 92.3% under the condition of $\varepsilon = 1.0$, and the re-recognition rate dropped to 6.7%, which was better than HE-only (15.8%) and DP-only (9.4%) scheme. While ensuring privacy, it achieves a balance between security and efficiency.
(3) In the aggregation statistics task, Take into account both accuracy and communication cost: On the CERT v6.2 data set, the framework maintains an average error of 1.92% under different client scales, and the communication overhead is much lower than that of the HE-only scheme. The stability and convergence are better than those of DP + Noise and other schemes, showing good scalability and robustness.
(4) It has wide applicability and practical deployment value: The proposed framework does not rely on hardware trust environments such as TEE and is suitable for various cloud computing tasks such as log aggregation, behaviour modelling, and access control. Comparative experiments prove that the convergence mechanism achieves an excellent trade-off among security, usability and resource overhead and has practical feasibility and popularization potential.

Although the proposed framework demonstrates enormous potential, there are still certain limitations. Firstly, the noise calibration mechanism assumes static sensitivity parameters, which may not be well adapted to highly dynamic data streams. Secondly, although homomorphic encryption can protect privacy, it still incurs relatively high computational overhead in large-scale real-time applications. Thirdly, the current prototype lacks formal proof under a composable privacy model, which may be necessary for high security critical deployment.

Future research will focus on the following directions: developing a real-time privacy budget allocation mechanism that can dynamically respond

based on query context and user preferences; Extend the current framework to federated learning scenarios to enhance privacy protection capabilities in distributed environments; Conduct formal security analysis for adaptive and inferential opponents to enhance system robustness; And optimize system latency and computational efficiency to support real-time data analysis requirements under edge cloud collaborative architecture.

Fundings

This study was funded by the project of the key research topic of Anhui Vocational and Adult Education Association (No. Azcj202104)

References

[1] Luo, W., Lv, Z., Lai, C., and Yang, T. "Efficient and secure cross-domain data sharing scheme with traceability for Industrial Internet," Computer Networks, vol. 260, pp. 111117, 2025.
[2] Guo, J., and Wang, L. "Learning to upgrade internet information security and protection strategy in big data era," Computer Communications, vol. 160, pp. 150–157, 2020.
[3] Ameur, Y., and Bouzefrane, S. "Enhancing privacy in VANETs through homomorphic encryption in machine learning applications," Procedia Computer Science, vol. 238, pp. 151–158, 2024.
[4] Alabdulatif, A. "GuardianAI: Privacy-preserving federated anomaly detection with differential privacy," Array, vol. 26, pp. 100381, 2025.
[5] Hasan, M. M., and Rahman, M. M. "Privacy-preserving polyp segmentation using federated learning with differential privacy," Smart Health, vol. 36, pp. 100551, 2025.
[6] Ci, S., Hu, S., Guan, D., and Koç, Ç. K. "Privacy-preserving word vectors learning using partially homomorphic encryption," Journal of Information Security and Applications, vol. 89, pp. 103999, 2025.
[7] Mittal, S. "Fully homomorphic encryption-based optimal key encryption for privacy preservation in the cloud sector," Journal of Information Security and Applications, vol. 91, pp. 104048, 2025.
[8] Naresh, V. S., and Ayyappa, D. "Enhancing security in software defined networks: Privacy-preserving intrusion detection with Homomorphic Encryption," Journal of Information Security and Applications, vol. 92, pp. 104084, 2025.

[9] R, P., and P, S. "Side-channel attack resilient implementation of homomorphic encryption using elliptic curve cryptography for secure cloud computing," Integration, vol. 104, pp. 102439, 2025.

[10] Tang, Z.-A., Duan, X.-F., Liang, R.-H., and Ding, Y. "Efficient multi-party privacy preserving federated k-means based on homomorphic encryption," Information Sciences, vol. 717, pp. 122335, 2025.

[11] Liu, W., You, L., Shao, Y., Shen, X., Hu, G., Shi, J., and Gao, S. "From accuracy to approximation: A survey on approximate homomorphic encryption and its applications," Computer Science Review, vol. 55, pp. 100689, 2025.

[12] Lv, L., Xiong, L., and Li, F. "PCPHE: A privacy comparison protocol for vulnerability detection based on homomorphic encryption," Journal of Information Security and Applications, vol. 84, pp. 103805, 2024.

[13] Mahato, G. K., Banerjee, A., Chakraborty, S. K., and Gao, X.-Z. "Privacy preserving verifiable federated learning scheme using blockchain and homomorphic encryption," Applied Soft Computing, vol. 167, pp. 112405, 2024.

[14] Ming, Y., Wang, S., Wang, C., Liu, H., Deng, Y., Zhao, Y., and Feng, J. "VCSA: Verifiable and collusion-resistant secure aggregation for federated learning using symmetric homomorphic encryption," Journal of Systems Architecture, vol. 156, pp. 103279, 2024.

[15] Mittal, S. "Fully homomorphic encryption-based optimal key encryption for privacy preservation in the cloud sector," Journal of Information Security and Applications, vol. 91, pp. 104048, 2025.

[16] Mohammed, M. A., and Wahab, H. B. A. "Enhancing IoT Data Security with Lightweight Blockchain and Okamoto Uchiyama Homomorphic Encryption," CMES – Computer Modeling in Engineering and Sciences, vol. 138, no. 2, pp. 1731–1748, 2023.

[17] Sun, H., Chen, X., and Yuan, K. "FL-EASGD: Federated Learning Privacy Security Method Based on Homomorphic Encryption," Computers, Materials and Continua, vol. 79, no. 2, pp. 2361–2373, 2024.

[18] Wu, X., Wang, J., and Zhang, T. "Integrating fully homomorphic encryption to enhance the security of blockchain applications," Future Generation Computer Systems, vol. 161, pp. 467–477, 2024.

[19] Xiong, R., Ren, W., Zhao, S., He, J., Ren, Y., Choo, K.-K. R., and Min, G. "CoPiFL: A collusion-resistant and privacy-preserving federated learning crowdsourcing scheme using blockchain and homomorphic encryption," Future Generation Computer Systems, vol. 156, pp. 95–104, 2024.

[20] Zhang, C., Zhang, X., Yang, X., Liu, B., Zhang, Y., and Zhou, R. "Poisoning attacks resilient privacy-preserving federated learning scheme based on lightweight homomorphic encryption," Information Fusion, vol. 121, pp. 103131, 2025.

[21] Zhou, F., Sun, J., Wang, Q., Zhang, Y., Hou, R., and Wang, C. "Efficient private information retrievals for single-server based on verifiable homomorphic encryption," Computer Standards & Interfaces, vol. 93, pp. 103961, 2025.

[22] He, C., and Ding, C. H. Q. "SecRASP: Next generation web application security protection methodology and framework," Computers & Security, vol. 154, pp. 104445, 2025.

[23] Kumar, A. S., and Revathy, S. "A hybrid soft computing with big data analytics based protection and recovery strategy for security enhancement in large scale real world online social networks," Theoretical Computer Science, vol. 927, pp. 15–30, 2022.

[24] Pleger, L. E., Guirguis, K., and Mertes, A. "Making public concerns tangible: An empirical study of German and UK citizens' perception of data protection and data security," Computers in Human Behavior, vol. 122, pp. 106830, 2021.

[25] Song, X., Xu, G., Huang, Y., and Dong, J. "DID-HVC-based Web3 healthcare data security and privacy protection scheme," Future Generation Computer Systems, vol. 158, pp. 267–276, 2024.

[26] Wang, F., Gai, Y., and Zhang, H. "Blockchain user digital identity big data and information security process protection based on network trust," Journal of King Saud University – Computer and Information Sciences, vol. 36, no. 4, pp. 102031, 2024.

[27] Xia, W., Zhang, L., Guo, Y., Zhang, H., and Cheng, L. "P4NSA: P4-based security protection technology for IPv6 neighbor solicitation and advertisement spoofing," Computers & Security, vol. 153, pp. 104400, 2025.

[28] Yang, M., Guo, J., Zhao, Z., Xu, T., and Bai, L. "Teenager Health Oriented Data Security and Privacy Protection Research for Smart Wearable Device," Procedia Computer Science, vol. 174, pp. 333–339, 2020.

[29] Ye, P.-G., Li, Z., Yang, Z., Chen, P., Zhang, Z., Li, N., and Zheng, J. "Periodic watermarking for copyright protection of large language models in cloud computing security," Computer Standards & Interfaces, vol. 94, pp. 103983, 2025.

[30] Zhang, C., Pan, Z., and Hou, C. "Marketing data security and privacy protection based on federated gamma in cloud computing environment," International Journal of Intelligent Networks, vol. 4, pp. 261–271, 2023.

Biography

Huang Yongsheng obtained a Bachelor's degree in Computer Science and Education from Anhui Normal University in 1999. He obtained a Master's degree in Computer Technology from Anhui University in 2013. Currently, he serves as the director and associate professor at the Modern Education Technology Center of Hefei Technology College. His areas of interest are smart campuses, computer networks, artificial intelligence, and big data.