# HAPB-FL: A Federated Learning and Hierarchical Key Agreement Framework with Adaptive Privacy Budgeting for Privacy Preservation in IoT

Juan Liu*, Yujing Zhang
Intelligent Manufacturing College, Shandong Polytechnic, Ji'nan 250104, China
E-mail: 6juan@163.com, zhangmere2025@163.com
*Corresponding author

**Keywords:** federal learning, privacy protection, privacy budget allocation, internet of things, hierarchical logical key hierarchy

**Received:** August 12, 2025

*Aiming at the problems of low efficiency, insufficient security and module fragmentation in the existing privacy protection schemes for the Internet of Things, a new privacy protection model is proposed. By introducing a data sensitivity weighting strategy through adaptive privacy budget allocation, the privacy budget and noise scale are dynamically adjusted. The cross-layer synchronization of root keys through hierarchical logical key tree subgroups and a decentralized architecture adapts to the low computing power requirements of edge devices. These two components simultaneously form a closed-loop synergy with federated learning. The experimental results show that the accuracy of the optimized federated algorithm gradually stabilizes after 100 rounds of communication and reaches 92.5±1.5%. The maximum recognition accuracy, predicted recall rate, and the harmonic mean of the recall rate reach 96.35%, 95.88%, and 96.10% respectively. The evaluation of the fusion model revealed that the overall coincidence degree of the output data of this model with the original data trajectory reached 97.6%. The average training time and the processing time of a single piece of data reached 1.2±0.2 min and 2.3±0.2 ms, respectively. The resource occupation ratio of this model reached a maximum of 51.3%. The above results indicate that the privacy protection model proposed by the research institute meets the requirements of large-scale privacy data processing and performs well in terms of real-time performance and efficiency.*

*Povzetek: Novi model varovanja zasebnosti za IoT omogoča učinkovito, varno in natančno obdelavo podatkov z visoko realnočasovno zmogljivostjo.*

## 1 Introduction

In recent years, with the widespread application of the Internet of Things (IoT), users have become increasingly aware of privacy protection and are paying more attention to the security of personal information in the IoT environment. Both individual and enterprise users have a strong demand for privacy protection products and services. The application of intelligent algorithms in the field of privacy protection not only enhances data encryption and security but also accurately implements privacy protection strategies [1]. For instance, there is a risk of privacy leakage in the current privacy protection of IoT. Most IoT devices are edge nodes, and the local model gradients they upload are vulnerable to reverse inference from the original data by attackers using "gradient inversion" technology. As one of the most transformative technologies in recent years, artificial intelligence algorithms are widely used in privacy protection [2]. Currently, mainstream privacy-preserving algorithms include differential privacy, homomorphic encryption, and Bayesian networks. However, these methods still face limitations such as difficulty in privacy budget allocation, complex key management, and low computational efficiency. Compared with similar algorithms, Federated Learning (FL) allows each participant to train the model locally using their own data, which fundamentally avoids sharing the original data. Since most computing tasks are performed on local devices or servers, computing resources are utilized more efficiently [3–4]. At the same time, the introduction of the Hierarchical Logical Key Hierarchy (HLKH) can effectively address the issue of key management in large-scale dynamic groups [5]. Therefore, this paper introduces a privacy-preserving model that integrates FL and HLKH. The research sets three core goals: ① Privacy protection strength meets the standard - ensuring that the model satisfies ε -differential privacy and resists attacks such as gradient inversion; ② Balance between efficiency and performance – under the premise of ensuring privacy, the model's accuracy rate is slightly lower than the baseline FL. The processing time for a single piece of data is short, the total resource occupancy is low, and the model adapts to the computing power limitations of edge devices in IoT. ③ Dynamic group adaptation – supports dynamic group management for over 100 nodes. When a node fails, the key update delay is low, and the accuracy fluctuation during device "join-exit" is small.

Table 1: Performance comparison analysis.

| Method | Core technology | Accuracy | Computational overhead | Privacy guarantee | Resource utilization rate |
|---|---|---|---|---|---|
| Distributed FL framework | Federated learning | 86.2% | 4.2 min for 100 training rounds | Vulnerable to gradient inversion attacks | 78.6% |
| Decentralized FL | Decentralized model aggregation | 82.3% | 3.8 min for 100 training rounds | No dynamic key management | 75.2% |
| FL ensemble framework | Federated learning + ensemble learning | 85.6% | 3.5 min for 100 training rounds | Gradient leakage risks | 72.8% |

## 2 Related works

FL, as a distributed machine learning framework, allows multiple data owners to collaboratively train a global model without sharing their original data. This mechanism, in which data remain usable but not visible, strikes a balance between privacy protection and model performance and attracted extensive attention from researchers worldwide. For example, Wen introduced a method to address the data island issue during collaborative model building by deploying the FL framework in real-world applications. Experimental results showed that as the FL model was applied in practice, performance bottlenecks emerged during training, which affected the efficiency and accuracy of FL models in real scenarios [6]. Beltrán's team introduced a decentralized FL approach to address issues caused by centralized global models, such as increased latency and system vulnerability. The experiment showed that this approach reduced the model's dependency on centralized architecture by 97.6% through decentralized aggregation [7]. Thomas and Myakala proposed an advanced FL integration framework to solve the limitations of cloud-based architecture and the need for decentralized alternatives. According to experimental results, the proposed framework improved model accuracy by 10% to 15% in heterogeneous environments and reduced communication costs by 25% [8]. Chen addressed the problem of data leakage by unauthorized entities and proposed a privacy-preserving computing method based on FL. Experimental results indicated that the method effectively protected user data privacy and identified several interesting directions for future research [9].

With the growing popularity of network applications, public awareness of privacy protection continues to increase, and many researchers around the world conduct in-depth studies in this area. For instance, Li et al. introduced a blockchain-based solution to resolve the conflict between data sharing and privacy protection in civil aviation enterprises. The experiment showed that the solution achieved both secure sharing and privacy protection of flight operation data, while also improving data sharing efficiency [10]. Huang et al. addressed security and privacy risks faced by users in the metaverse by proposing a new form of cyberspace divided into four economic domains. Based on these domains and current findings, the study optimized security and privacy concerns in the metaverse. The experimental results showed that this structure reduced privacy leakage risks [11]. Rodriguez E proposed a privacy protection scheme for data in IoT devices based on machine learning and deep learning. Experimental results show that this scheme can provide effective protection against different threats and attacks [12]. Das et al. focused on identity information leakage attacks in large language models and introduced a potential defense mechanism. Experiments showed that this mechanism guided future research directions in the field [13]. To further quantify the performance gap and limitations of the existing methods, a comparative analysis of the core indicators of the above three mainstream studies was conducted, as shown in Table 1.

In summary, current studies have made progress in privacy protection, but problems such as low efficiency and limited security in data protection still exist. FL can be integrated with a hierarchical dynamic group key agreement protocol to address these challenges. Therefore, the privacy-preserving model introduced in this study demonstrates practical value and is expected to improve the efficiency and comprehensiveness of privacy protection.

## 3 Privacy protection strategy based on the integration of APB-FL and HLKH

### 3.1 Privacy algorithm optimization based on FL

With the rapid development of artificial intelligence and the widespread use of mobile IoT devices, the explosive growth of data promotes service optimization while also posing challenges in data processing and privacy protection [14]. Currently, data are scattered across institutions, constrained by privacy regulations and business barriers, leading to data silos [15–16]. To address this issue, the study applies FL to integrate model updates from multiple parties while preserving data value and avoiding privacy leakage. The FL system structure is shown in Figure 1.

As shown in Figure 1, the global server first completes the model initialization and then selects clients based on multiple dimensions such as device resources, data quality, and remaining privacy budget. Subsequently, the global model is distributed to the selected clients through hierarchical encryption. Each client independently trains based on the local heterogeneous dataset, generates model parameters, and then encrypts and uploads them after signature verification. The central server updates the aggregation model based on the "weighted local sample size of the client", and simultaneously evaluates the model from both data performance and system performance dimensions. If it does not converge, the process of "client selection – encrypted distribution – local training – signature upload – aggregation evaluation" is repeated and iterated until the model converges or reaches the preset number of rounds. Clients train locally based on the global model to minimize local loss, as shown in Equation (1).

$$L_i(\omega)=\frac{1}{n_i}\sum_{(x,y)\in D_i} l\left(f\left(x_j,\omega,\psi\right),y_j\right) \quad (1)$$
$$+\lambda\cdot\Omega(\omega)$$

In Equation (1), $L_i$ denotes the local loss value of the $i$-th client. $n_i$ represents the local IoT dataset of the $i$-th client. $\omega$ denotes global model parameters (including weights, biases, and other trainable parameters). $i$ stands for the client index. $D_i$ indicates the local dataset of the client (containing raw data collected by IoT devices). $l(*)$ represents the cross-entropy loss function (tailored for IoT classification tasks and serves as the core component of local losses). $x_j$ is the $j$-th sample in $D_i$, $y_j$ is the true label corresponding to $x_j$, $\lambda$ is the regularization coefficient (set to 0.001 to prevent overfitting in small-sample IoT scenarios), and $\Omega(\omega)$ denotes the regularization term. The server collects local model from all clients and performs a weighted average operation based on data volume as shown in Equation (2).

$$\omega_{t+1}^{i}=\sum_{i=1}^{K}\frac{n_i}{N}\omega_t^i \quad (2)$$

In Equation (2), $\omega_{t+1}^{i}$ denotes the aggregated weight of the $t+1$-th client after $i$ iterations, $K$ represents the number of clients, $N$ stands for total data volume, and $t$ is the index of federated training rounds. The objective of federated learning is to minimize the joint loss across all client data, with the calculation process of the joint loss function illustrated in Equation (3).

$$L_{global}(\omega)=\sum_{i=1}^{K}\frac{n_i}{N}L_i(\omega)=$$
$$\frac{1}{N}\sum_{i=1}^{K}\sum_{(x,y)\in D_i} l\left(f\left(x_j,\omega,\psi\right),y_j\right) \quad (3)$$
$$+\lambda\cdot\Omega(\omega)$$

In Equation (3), the weight $\frac{n_i}{N}$ ensures that the global model has a greater impact on the client with larger data volume. Equation (3) shows that the joint loss strongly correlates with the local losses. Although FL enables reliable model training, it still faces limitations such as uneven privacy protection, decreased accuracy due to noise, and inefficiencies caused by data heterogeneity. Therefore, this study introduces the Adaptive Privacy Budget Allocation (APBA) mechanism to address these issues. APBA protects sensitive data and balances privacy and accuracy through feedback. Its workflow is illustrated in Figure 2.

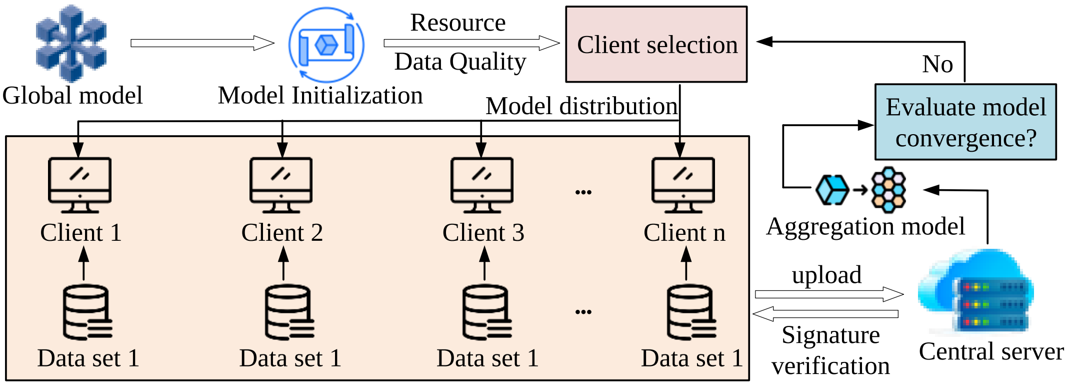


Figure 1: FL system structure diagram.

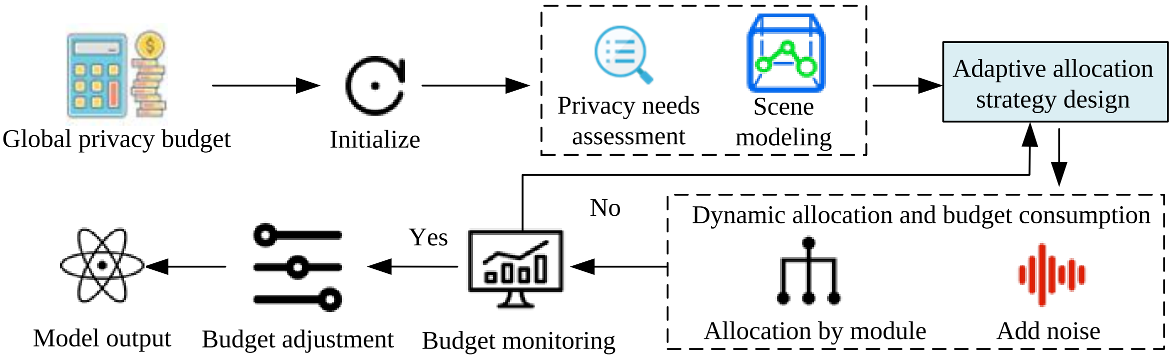


Figure 2: Schematic diagram of APBA workflow.

As shown in Figure 2, APBA first initializes the global privacy budget, then assesses privacy requirements and models the scenarios - the assessment process classifies privacy levels based on the types of IoT devices, such as medical monitoring devices and environmental perception sensors. Medical devices have high privacy requirements, so a smaller single-round budget is allocated, while environmental sensors have relatively low privacy requirements, and the budget can be appropriately relaxed. At the same time, the L1 global sensitivity model is adopted to quantify data sensitivity and adapt to the high-dimensional sparse data features of the IoT. In the scene modeling stage, it is clearly assumed that the data distribution is non-independent identically distributed (Non-IID), and the Dirichlet distribution (parameter α=0.5) is adopted to simulate the unbalanced characteristics of data categories from different clients, which is in line with the actual data collection differences of IoT devices. Then, based on the hierarchical strategy, the allocation strategy is formulated, and corresponding noise is added for dynamic allocation and budget consumption prediction. Finally, monitor the budget requirements. If optimization is needed, loop back to the strategy design steps. If met, adjust and output the target model. When $m$ differential privacy mechanisms are sequentially applied, the total budget must satisfy the condition in Equation (4).

$$\varepsilon_{total} \le \sum\nolimits_{g=1}^{m} \varepsilon_g \qquad (4)$$

In Equation (4), $\varepsilon_g$ represents the privacy budget for the $g$ -th mechanism. This is used in multi-step operations such as gradient uploads and server aggregation in FL. If $m$ mechanisms are applied independently to disjoint datasets, the total budget is calculated as in Equation (5).

$$\varepsilon_{total} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m\} \qquad (5)$$

Equation (5) represents a parallel composition, suitable for multiple clients processing non-overlapping datasets. During aggregation, APBA adds Laplace and Gaussian noise to gradients or model parameters. The relationship between privacy budget, standard deviation, and noise scale is shown in Equation (6).

$$\begin{cases} \sigma \ge \dfrac{\Delta f}{\varepsilon} \square \sqrt{\dfrac{2\ln 1.25}{\varepsilon}} \\ b = \dfrac{\Delta f'}{\varepsilon} \end{cases} \qquad (6)$$

In Equation (6), $\Delta f'$ represents the L1 global sensitivity of the mechanism, $\varepsilon$ represents the discrete sensitivity, and $\Delta f$ represents the local differential privacy budget for a single round of training. It needs to be dynamically determined in combination with the privacy requirements of the dataset: In the Synthetic dataset (simulating environmental sensor data with moderate privacy requirements) $\varepsilon = 0.08$, the CIFAR10 dataset (simulating visual monitoring data with relatively high privacy requirements) $\varepsilon = 0.06$ , the MovieLens dataset (user behavior data with moderate privacy requirements) $\varepsilon = 0.07$ , and the Adult dataset (user attribute data with high privacy requirements) $\varepsilon = 0.05$; The probability of privacy protection failure is uniformly set at 10-5. Quantify the degree of privacy leakage through Renyi Differential Privacy (RDP), the differential privacy protection of local model update is realized by using the Laplace noise mechanism, and the noise scale $b$ is determined jointly by the "discrete sensitivity $\Delta f'$" and the "allocated privacy budget $\varepsilon$". When the used budget $\varepsilon_{used}$ exceeds or equals the total budget $\varepsilon_{total}$ , the process terminates. The computation of the used budget $\varepsilon_{used}$ is shown in Equation (7).

$$\varepsilon_{used} = \sum\nolimits_{t=1}^{T} \sum\nolimits_{i=1}^{n} \varepsilon_i^{(t)} \qquad (7)$$

In Equation (7), $T$ is the training round, and $\varepsilon_i^{(t)}$ is the budget used by client $i$ in round $t$ . The study names the personalized FL with APBA as APB-FL, which dynamically allocates privacy budgets by identifying sensitive parameters and stages. The APB-FL framework is shown in Figure 3.

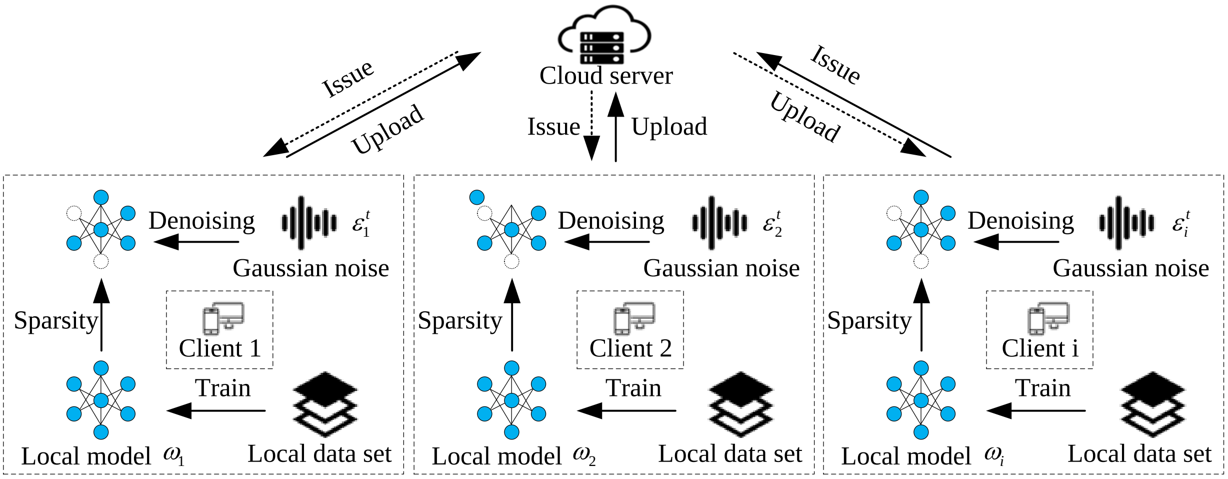


Figure 3: APB-FL framework diagram.

As shown in Figure 3, APB-FL consists of three steps. First, clients train local models on their datasets. Second, model sparsification reduces parameters and noise is added to protect sensitive data, lowering budget use for regular clients and improving aggregation. Third, clients contributing more data or participating in more rounds are allocated budgets based on risk. Clients update local models using gradient descent, as shown in Equation (8).

$$\theta_i^{t+1} = \theta_i^t - \eta \cdot \nabla L\left(\theta_i^t ; D_i\right) + \mathrm{N}\left(0, \sigma^2\right) \qquad (8)$$

In Equation (8), $\eta$ is the learning rate, $\nabla L\left(\theta_i^t ; D_i\right)$ is the gradient on dataset $D_i$, and $\mathrm{N}\left(0, \sigma^2\right)$ is the Gaussian noise added. The server collects local parameters and performs weighted aggregation to get the global model, as shown in Equation (9).

$$\theta^{t+1} = \sum_{i=1}^{N} \omega_i \theta_i^{t+1} \qquad (9)$$

In Equation (9), $\omega_i$ is the weight of client $i$. The weights of different clients are usually different and are generally determined based on factors such as the amount of data on the client.

## 3.2 Privacy protection model design integrating APB-FL and HLKH

While APB-FL mitigates data silos with privacy protection, it still faces issues such as overfitting, unclear responsibility for data leakage, and network congestion due to frequent gradient uploads in IoT devices. To solve these, this study applies a hierarchical dynamic group key agreement protocol. This approach reduces communication and computation costs, enhances system robustness via dynamic key management, and resists single-point failures using a decentralized structure [17–18]. A well-designed hierarchical structure is crucial, and the IoT hierarchical structure is shown in Figure 4.

As shown in Figure 4, the layered IoT strategy divides the entire architecture into four layers. The device layer collects data from IoT devices and performs basic preprocessing. The preprocessed data are then transmitted to the edge layer, where edge computing is conducted to carry out local decision-making and control tasks. The data from the edge layer are passed to the network layer, which first adapts communication protocols to ensure smooth data transmission. Afterward, the processed data reach the application layer for business logic operations, where information extraction and analysis are performed. The sensitivity-based budget allocation is computed as shown in Equation (10).

$$\varepsilon_k = \varepsilon_{total} \cdot \frac{S_k}{\sum_{k=1}^{K} S_k} \qquad (10)$$

In Equation (10), $S_k$ represents the data sensitivity of a cluster. HLKH divides the group into multiple subgroups through a hierarchical structure, where each subgroup manages its own keys independently. Inter-group communication is realized by aggregating keys at the upper layer. This structure specifically addresses the key management challenges in large-scale dynamic groups. Therefore, this study introduces the HLKH protocol to optimize the algorithm. The framework of HLKH is shown in Figure 5.

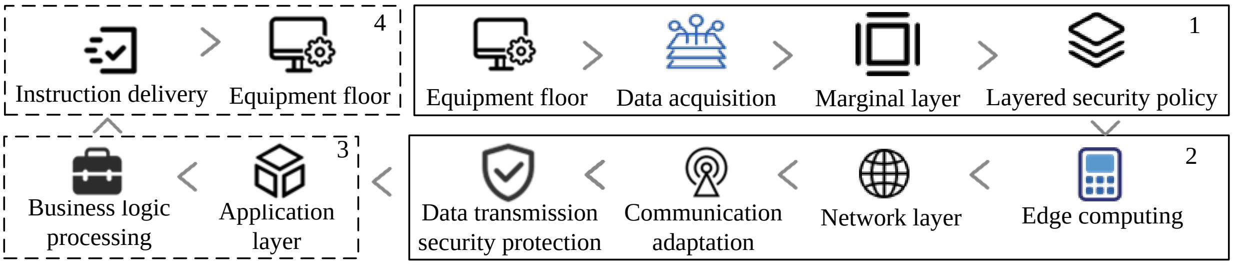


Figure 4: Hierarchical strategy structure diagram of the IoT.

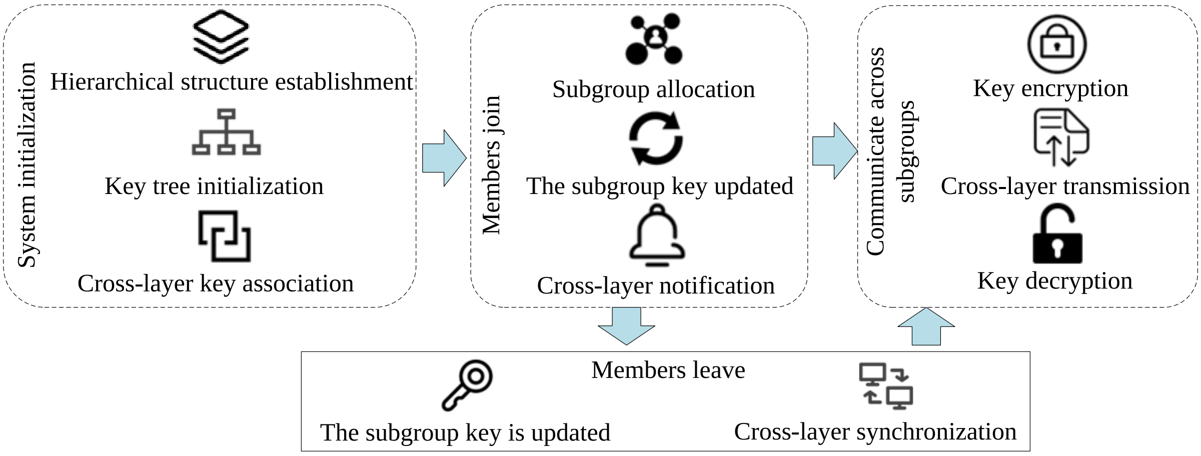


Figure 5: HLKH protocol structure diagram.

As shown in Figure 5, the HLKH protocol begins with system initialization. This step includes the establishment of the hierarchy, the construction of logical key trees within each subgroup, and the association of inter-layer keys by binding the upper-layer aggregated key to the root key of the lower subgroup using encryption algorithms. The second step assigns subgroups, updates the key trees, and encrypts the new subgroup root key with the upper-layer key before transmitting it through a secure channel to other subgroup managers. The final step involves encryption, cross-layer transmission, and decryption of keys. The root key derivation process from child nodes is shown in Equation (11).

$$GK = H\left(SK_{left} \| SK_{right} \| timestap\right) \qquad (11)$$

In Equation (11), $H$ is explicitly specified as the SHA-256 hash function, which has the following security properties: 1. Collision resistance - It is difficult to find two different inputs that produce the same hash value, which can prevent attackers from forging child node keys to derive the root key. 2. Pre-image attack - The original input cannot be inferred from the hash result, ensuring the privacy of the child node key. 3. High efficiency - Low computational complexity, suitable for the limited computing resources of IoT edge devices, $SK_{left}$ and $SK_{right}$ represent the left and right child keys respectively, and $timestap$ is used to prevent replay attacks. Within each subgroup, members' private keys are updated upward along the key tree path, which only affects members within the same subgroup. The generation of a member's private key from a leaf node is illustrated in Equation (12).

$$MK_i = H\left(SK_{leaf} \| device\_ID_i\right) \qquad (12)$$

In Equation (12), $SK_{leaf}$ is the leaf node key, and $device\_ID_i$ is the unique identifier of the device. The sender encrypts the subgroup key using the upper-layer key to generate a cross-layer encryption key. The encryption process of the subgroup key is shown in Equation (13).

$$\begin{aligned} &Encrypted\_GK = \\ &AES-Encrypt\left(Super\_Key, GK\right) \end{aligned} \qquad (13)$$

In Equation (13), $AES-Encrypt$ is clearly specified as the AES-256-GCM symmetric encryption algorithm, and its security properties and applicability are as follows: 1. Confidentiality - The 256-bit key length can resist all currently known brute-force and differential attacks, meeting the privacy requirements of sensitive scenarios such as healthcare and finance in IoT. 2. Integrity and Authentication - The GCM mode, by attaching message authentication codes, can detect data tampering or forgery during transmission and prevent cross-layer keys from being maliciously replaced. 3. Parallel computing support - The encryption and authentication processes can be executed in parallel, reducing the communication latency of IoT devices, $Super\_Key$ is the upper-layer key, and symmetric encryption is used. The recipient subgroup manager uses the local key $Super\_Key$ to decrypt $Encrypted\_GK$ and retrieve $GK$, enabling inter-group communication. The key derivation function is computed as shown in Equation (14).

$$\begin{cases} PRK = HKDF-Extract\left(salt, IKM\right) \\ OKM = HKDF-Expand\left(PRK, \inf o, L\right) \end{cases} \qquad (14)$$

In Equation (14), $HKDF$ explicitly adopts SHA-256 as the underlying hash function, and its security properties include: 1. Forward security - If the subsequent derived key is leaked, it does not affect the security of the original input key material, which is suitable for the long-term deployment scenarios of IoT devices. 2. Key separation - Different context information can be derived into independent keys to prevent key leakage at one level from affecting other levels. 3. Lightweight implementation - No complex computation is required, it can run efficiently in resource-constrained IoT terminals such as sensors and embedded devices, $PRK$ is the pseudorandom key, $IKM$ is the input key material, $\inf o$ is the context information, and $L$ is the output key length. HLKH uses a hierarchical key tree structure and a dynamic update mechanism to achieve efficient key management in large-scale dynamic groups. Its core equations involve hash functions, symmetric encryption, and key derivation functions to ensure the security of key generation, update, and cross-layer transmission. The integrated privacy-preserving model that combines HLKH and APB-FL is named HAPB-FL, and its architecture is illustrated in Figure 6.

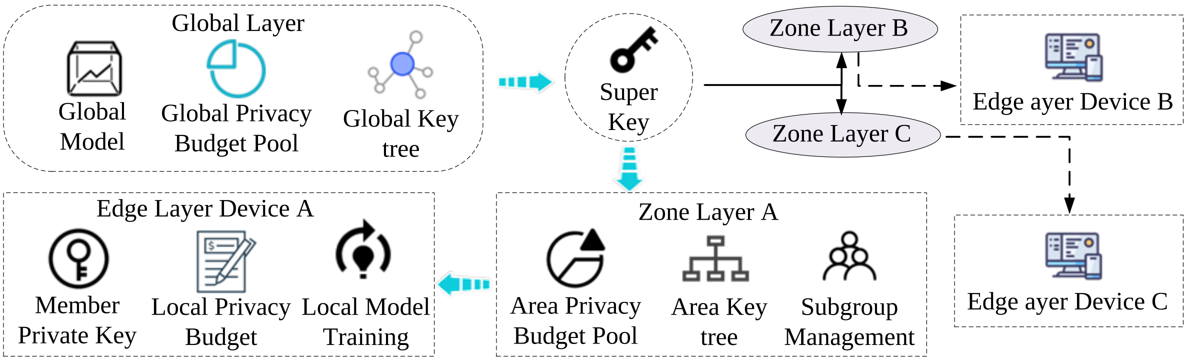
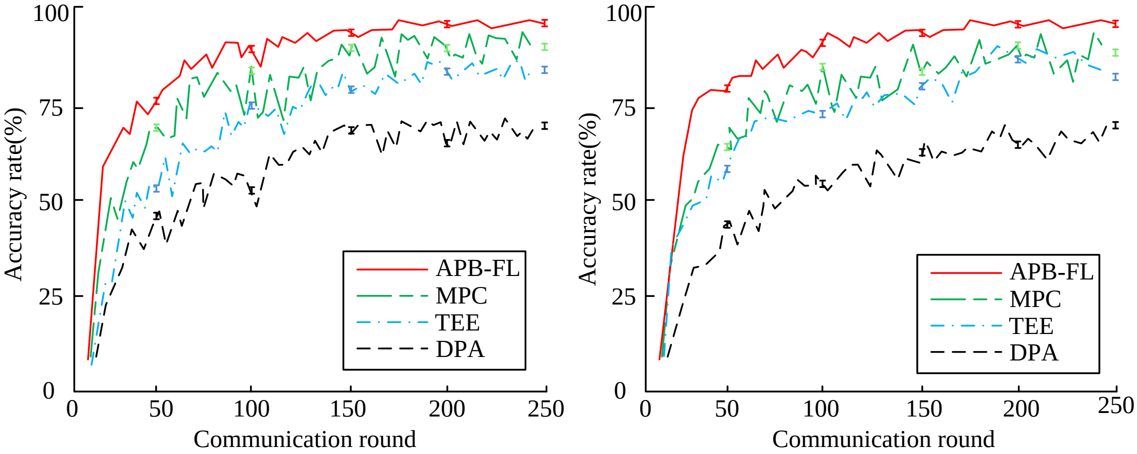


Figure 6: HAPB-FL privacy protection model framework diagram.

Table 2: Experimental hyperparameter settings.

| Parameter type | Value (Synthetic dataset) | Value (CIFAR10 dataset) |
|---|---|---|
| Batch size | 32 | 32 |
| Learning rate | 0.01 | 0.001 |
| Communication rounds | 250 | 250 |
| Local epochs | 5 | 5 |

Note: Based on the 95% confidence interval of three independent experiments



(a) Accuracy curve in the Synthetic dataset

(b) Accuracy curve in the CIFAR10 dataset

Figure 7: Accuracy training and testing results of different datasets.

As shown in Figure 6, HAPB-FL adopts a hierarchical architecture that includes three core modules: key management, privacy budget allocation, and FL training, as well as dynamic member management and secure communication mechanisms. The architecture divides the model into a global layer, a regional layer, and an edge layer, with specific processes defined for each part. The modules are connected by the HLKH key management module to form a hierarchical key chain, where each layer's key is derived from the lower layer. During local training, Laplace noise is added at the edge devices, consuming local privacy budgets. The structure clearly demonstrates how HAPB-FL achieves dynamic key management, APBA, and secure aggregation in FL, making it suitable for privacy-preserving applications in large-scale dynamic IoT groups.

# 4 Performance evaluation of the improved FL-based privacy protection method

## 4.1 Performance validation of the improved APB-FL

To verify the superiority of the APB-FL privacy-preserving algorithm, this study compared it with three other privacy-preserving methods: Secure Multi-Party Computation (MPC), Trusted Execution Environment (TEE), and Differential Privacy Algorithm (DPA). The experiments were conducted on a system running Windows 11 with Ubuntu 22.04 OS. The simulation was implemented using the FedML framework, with the Adam optimizer and Python 3.8. The hardware configuration included an NVIDIA GeForce RTX 4080 GPU and 64GB RAM. To ensure the reliability and authenticity of the experiments, the experimental dataset was selected from the CIFAR10 dataset and a Synthetic dataset constructed with reference to the Kaggle IoT anomaly detection dataset. This dataset supports 30 clients to participate in the training. Each round adopts the "random sampling + data volume threshold" strategy. The Synthetic dataset uses a 2-layer MLP. The input dimension is 5. Its label space adopts binary classification (normal/abnormal), with the proportion of abnormal samples being 15%. GAN is used to generate sensor data similar to real environments to simulate the distribution shift of the data. The CIFAR10 dataset adopts a lightweight CNN with a total parameter count of ≤2M. Both are implemented based on the FedML framework and are suitable for the lightweight requirements of IoT edge devices. The specific hyperparameter settings are shown in Table 2.

APB-FL, MPC, TEE, and DPA were respectively evaluated on both datasets for accuracy. The results are shown in Figure 7.

As shown in Figure 7(a), when trained on the Synthetic dataset, the accuracy of APB-FL reached 75.0±1.2% after 50 communication rounds. The overall accuracy curve gradually stabilized after 100 rounds and peaked at 92.5±1.5%. In contrast, the accuracy curve of MPC showed significant fluctuations, with the most dramatic changes occurring between rounds 50 and 100, where the highest accuracy reached 80.1±2.8%. TEE presented a similar trend to MPC, achieving a maximum accuracy of 77.6±2.4%. DPA yielded the lowest average accuracy among all algorithms, stabilizing around 65%. As shown in Figure 7(b), in training on the CIFAR10 dataset, APB-FL demonstrated faster convergence. It converged after 50 communication rounds, ultimately reaching 95.2±1.1%. The accuracy trends of MPC and DPA showed little change compared to the Synthetic dataset. TEE, however, exhibited an overall accuracy improvement of approximately 6%. To further explore the impact of client participation on model accuracy, the study conducted an accuracy evaluation under a total of 100 users, as shown in Figure 8.

According to Figure 8, when the number of participating clients increased from 20 to 40, APB-FL experienced a 6.5% drop in accuracy. However, as the number of clients increased to 100, its accuracy remained relatively stable, eventually stabilizing at 78.2%. MPC's accuracy dropped from 76.3% to 34.2%, while TEE's decreased from 73.6% to 28.6%. Although DPA showed a less dramatic decline compared to MPC and TEE, its initial accuracy was only 52.6% and ultimately fell to 26.9%. These results demonstrated that, compared with the other algorithms, APB-FL maintained higher accuracy even when handling large-scale client data. To further demonstrate the advantages of APB-FL, this study compared it with MPC, TEE, and DPA based on three evaluation metrics: precision, recall, and F1-score. The results are shown in Table 3.

As presented in Table 3, when tested on the Synthetic dataset, APB-FL achieved a precision of 96.35%, recall of 95.88%, and F1-score of 96.10%. MPC achieved a precision of 92.14% and a recall of 90.23%, which were 4.21% and 5.65% lower than those of APB-FL, respectively. When tested on the CIFAR10 dataset, TEE achieved a recall of 88.95% and an F1-score of 89.52%, both lower than those of APB-FL. For DPA, all three metrics remained below 90% in both datasets. Its recall was only 87.96%, which was 10.27% lower than that of APB-FL. These results confirmed the superior predictive performance of APB-FL in identifying private data.

## 4.2 Evaluation and analysis of the FL privacy protection model based on HLKH

After verifying the superiority of the APB-FL algorithm, the study further evaluated the performance of the improved FL privacy-preserving model based on the HLKH protocol. It was compared with privacy-preserving models constructed using MPC, TEE, and DPA. The experiments were conducted using PyTorch as the core deep learning framework, with Anaconda 3 as the development environment. Model training was performed in the MATLAB R2006 simulation environment, using the MovieLens and Adult datasets. To evaluate how well each model could reflect the actual data, this study compared the processed data against the original data for all four models in both datasets. The results are shown in Figure 9.

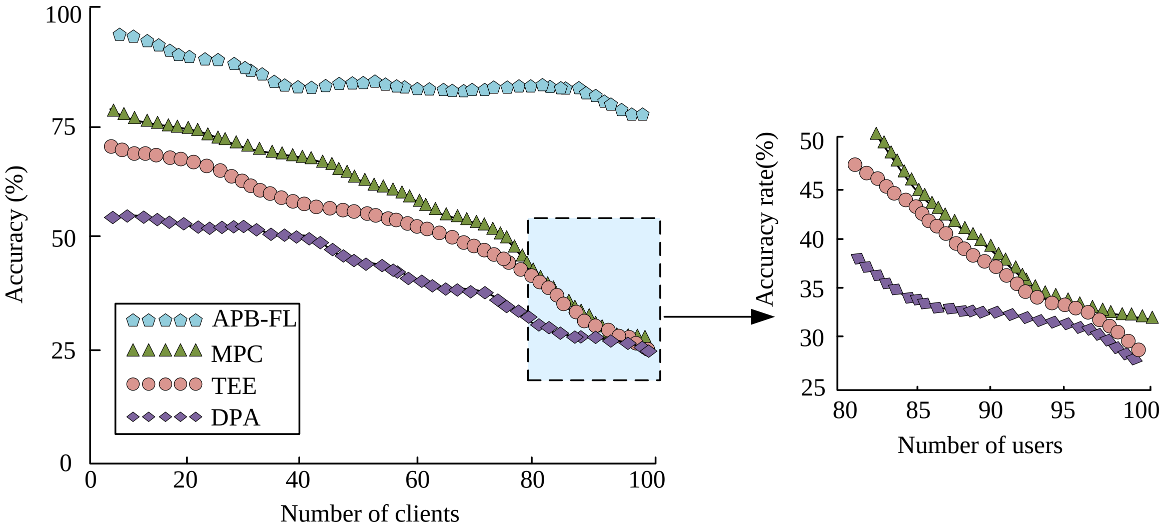
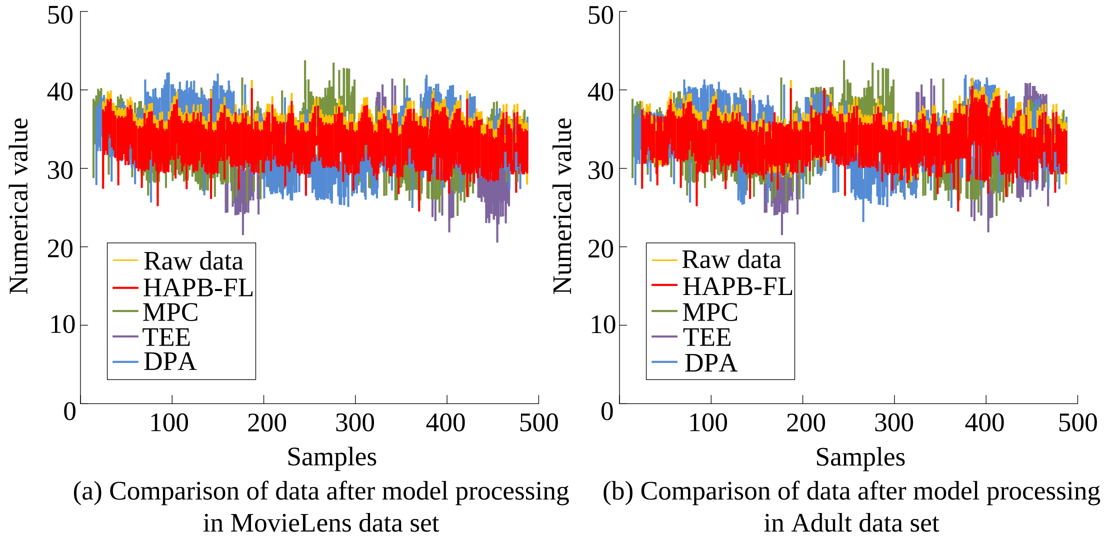


Figure 8: Accuracy experimental results under different numbers of clients.

Table 3: Test results of precision, recall and F1-score.

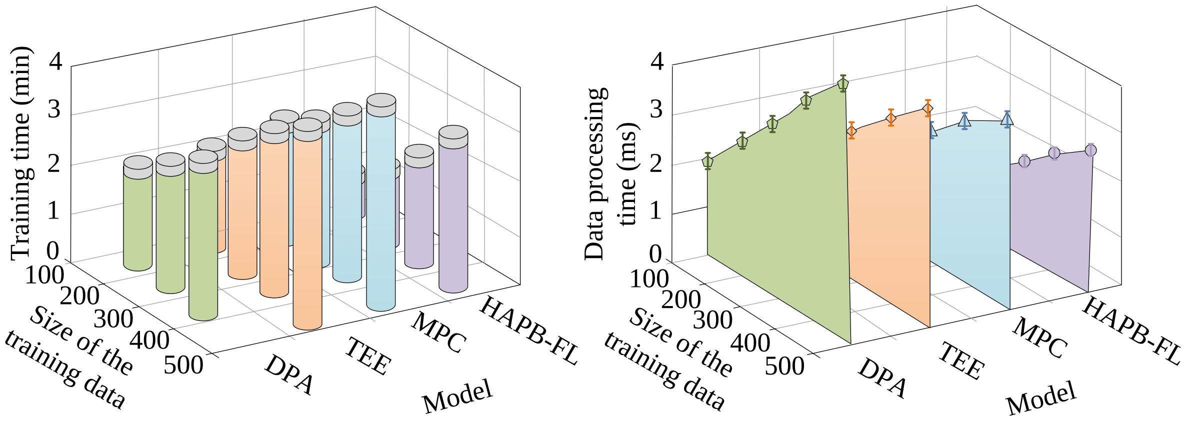
| Dataset | Algorithm | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|---|
| Synthetic | APB-FL | 96.35 | 95.88 | 96.10 |
| | MPC | 92.14 | 90.23 | 91.18 |
| | TEE | 91.66 | 90.25 | 90.95 |
| | DPA | 84.83 | 89.68 | 86.90 |
| CIFAR10 | APB-FL | 97.45 | 98.23 | 97.84 |
| | MPC | 89.23 | 90.66 | 89.94 |
| | TEE | 90.11 | 88.95 | 89.52 |
| | DPA | 85.98 | 87.96 | 86.96 |



(a) Comparison of data after model processing in MovieLens data set

(b) Comparison of data after model processing in Adult data set

Figure 9: Experimental results of closeness to original data.



(a) Compare the training time of the model

(b) Compare model data processing speed

Note: Based on the 95% confidence interval of four independent experiments

Figure 10: Comparison of training time and data processing speed.

As shown in Figure 9, the HAPB-FL model produced a smooth fluctuating curve ranging between values 40 and 30. The output data from HAPB-FL overlapped with the original data by 97.6%, indicating higher data authenticity. In contrast, MPC deviated significantly from the original data curve in samples numbered 100–200 and 300–400, and showed abrupt increases in the 200–300 range. TEE and DPA models showed even lower overlap, at 71.2% and 65.8%, respectively. These results indicated that the HAPB-FL model achieved a higher degree of statistical consistency with the original data, demonstrating better fitting performance than the compared models. To further assess computational efficiency, the study compared the training time and data processing speed of the four models, as shown in Figure 10. In the experiment, the network simulation is based on the 5G edge communication scenario, with a communication delay of 20±5ms and an uplink bandwidth of 10Mbps. The heterogeneity level of devices is reflected by the difference in computing power.

Figure 10(a) shows that the training time of all models increased as the training scale expanded. However, the HAPB-FL model required less time overall. It took only 1.2±0.2 minutes to complete training at a scale of 100, and 2.3±0.3 minutes at the maximum scale of 500. In comparison, MPC, TEE, and DPA required 3.5±0.5, 3.3±0.6, and 3.7±0.3 minutes respectively at the same scale. This shorter training time enhanced the model's practicality and scalability. As shown in Figure 10(b), under the maximum training scale of 500, HAPB-FL processed a single data sample in only 2.3±0.2ms, compared to 4ms for the other models. This significantly improved processing efficiency, The difference is statistically significant (t-test, $p$<0.01), making it suitable for scenarios requiring high real-time performance. To evaluate the resource consumption during model operation, the study compared the resource usage of all four models, as shown in Figure 11. Resource usage data is jointly monitored through the performance analysis tools Py-Spy (version 0.3.14) and nvidia-smi (version 535.104.05) - Py-Spy is used to calculate CPU, memory, and network usage rates (sampling frequency once per second). nvidia-smi is used to accurately obtain GPU memory and computing core occupancy rates. The experiment was repeated five times. The average resource occupation within 10 minutes of each run was taken to ensure data stability. The size of the experimental batches is uniformly set at 64 to avoid fluctuations in resource occupation caused by batch differences and ensure the fairness of the comparison.

As shown in Figure 11(a), during operation, the HAPB-FL model consumed 9.35% CPU, 12.35% GPU, 10.82% memory, 12.75% network bandwidth, and 6.03% in other resources, totaling 51.3% resource usage. In contrast, the total resource usage of the other models remained around 81%. These findings showed that the lower resource consumption of HAPB-FL enabled it to run on various hardware platforms, particularly in resource-constrained environments such as mobile devices and embedded systems. To verify the ability of the HAPB-FL model to resist single points of failure and support dynamic groups, the study added targeted experiments, focusing on the edge layer nodes in the hierarchical architecture of the HLKH protocol. Five gradients were set to simulate node failures of different degrees, such as a 20% failure, that is, 10 edge nodes suddenly going offline. Failures were triggered respectively in the 20th, 40th, and 60th rounds of federated training. To avoid result deviations caused by a single fault timing, the three key evaluation indicators are compared as shown in Table 4.

As shown in Table 4, even with 25% edge node failures, the HAPB-FL model maintains an accuracy rate exceeding 89%, with key update latency ≤32ms and data processing success rate surpassing 88%. The core reason lies in the decentralized hierarchical key architecture of the HLKH protocol: failed nodes only affect their respective subgroups, while other subgroups can continue communication through upper-layer aggregation keys, effectively preventing "global paralysis caused by single-point failures".To clarify the independent contributions of each component in the four schemes of baseline federated learning, FL+APBA, FL+HLKH, and full HAPB-FL, the study further designed ablation experiments. The evaluation indicators focused on model performance, privacy protection strength, computational efficiency, and resource consumption. The hyperparameters (batch size 64, learning rate 0.005, training rounds 80) and data preprocessing procedures of the four groups of schemes are exactly the same, with only the differentiated components being variables. The experimental results are shown in Table 5.

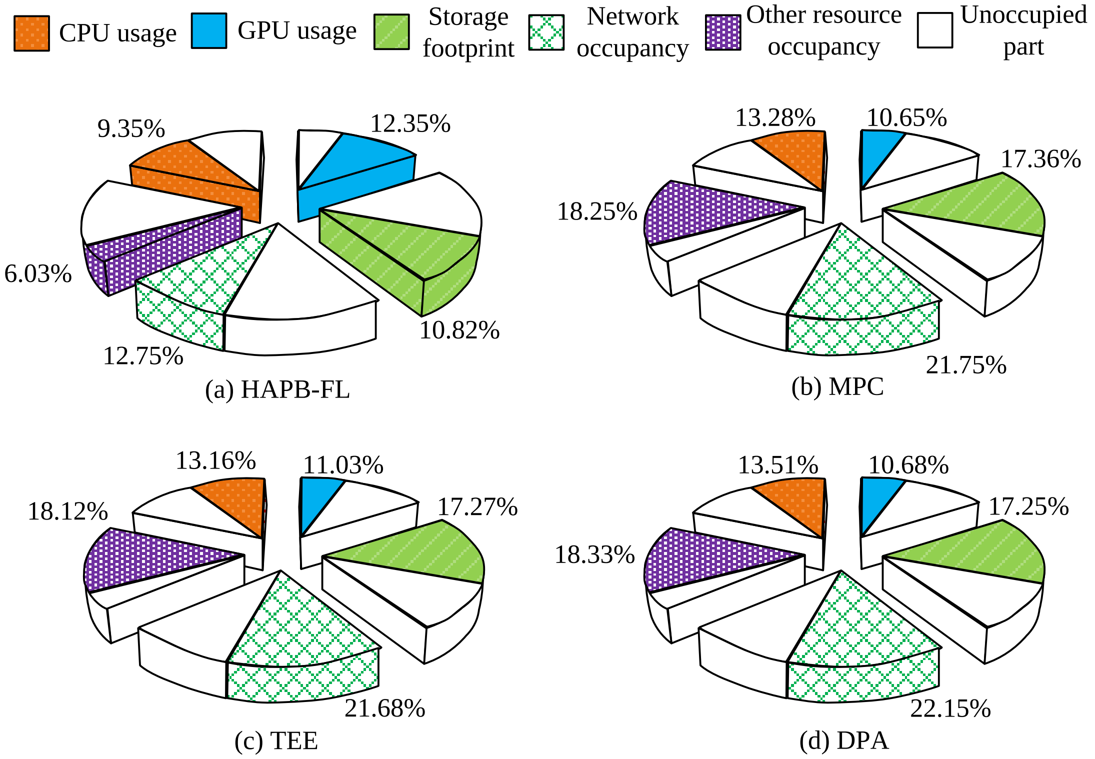


Figure 11: Resource usage experiment results.

Table 4: Simulation experiment results of node failures.

| Node Failure Rate | Accuracy Retention Rate (Failure at Round 40) | Key Update Latency | Data Processing Success Rate |
|---|---|---|---|
| 5% | 98.7% | 18ms | 99.2% |
| 10% | 96.5% | 22ms | 97.8% |
| 15% | 94.3% | 25ms | 95.1% |
| 20% | 92.1% | 28ms | 92.5% |
| 25% | 89.6% | 32ms | 88.7% |

Table 5: Results of the ablation experiment.

| Experimental Scheme | Accuracy (%) | Total Privacy Budget | Average Training Time (min) | Total Resource Occupancy Rate (%) |
|---|---|---|---|---|
| baseline FL | 86.2 | - (No Privacy Protection) | 3.5 | 78.6 |
| FL+APBA | 84.5 | 6.8 | 3.7 | 76.3 |
| FL+HLKH | 85.8 | - (No Privacy Protection) | 2.8 | 55.1 |
| HAPB-FL | 83.9 | 6.0 | 2.3 | 51.3 |

As shown in Table 5, the total privacy budget of HAPB-FL is the lowest among the four combination schemes. The training time (2.3min) and resource occupancy rate (51.3%) also reach the lowest. The accuracy rate only decreases by 2.9% compared with the baseline FL, achieving a balance of "privacy - performance - efficiency" in the IoT scenario.

### 4.3 Security analysis and threat model of privacy protection model

Attacker definition: Adopting a "semi-honest - malicious hybrid model" - semi-honest attackers follow the protocol but steal data, while malicious attackers tamper with data/keys; The core objective is to reverse-engineer the original data, contaminate the global model, and crack the hierarchical key. System assumption: Global servers and ≥50% of edge nodes/terminals are trusted entities; The key transmission is guaranteed by TLS 1.3. The local environment of IoT (iot) devices is "weakly trusted" (with only the risk of data leakage and no hardware tampering), which is consistent with the local training scenario of APB-FL.

①Inference attack: Reverse-engineer the original data through the gradient uploaded by the client. Defense: Relying on the APBA mechanism, privacy budgets are dynamically allocated based on data sensitivity. More Laplacian noise is injected into highly sensitive clients to reduce the correlation between the gradient and the original data. Combined with the sensitivity calibration of 10 data clusters, enhance the privacy protection of high-dimensional data. ②Gradient poisoning attack: Malicious clients upload and tamper with gradients, polluting the global model. Defense: Based on the HLKH protocol, the client needs to generate a "device-key binding identity identifier" through Equation 12, and the server only accepts legal gradients. At the same time, filter out outliers of the gradient and eliminate abnormal updates. ③Key cracking and cross-layer attacks: Cracking HLKH keys or leaking single group keys to affect the overall situation. Defense: Equation 11 uses SHA-256 and Equation 13 uses AES-256-GCM to ensure key security; The upper-layer aggregation key is dynamically updated every 10 rounds through 14 HKDF to limit the scope of leakage impact. Key update requires signature verification from three trusted edge nodes.

## 5 Discussion

The HAPB-FL model proposed in the research, in terms of privacy-accuracy balance, has an accuracy rate of 92.5±1.5% on the Synthetic dataset and 95.2±1.1% on the CIFAR10 dataset, which is superior to the traditional schemes MPC, DPA, and the integrated AIoT architecture model proposed by Sun P et al. in 2024. This is attributed to the sensitivity weighting of APBA for noise processing [19]. In terms of computational efficiency, at a training scale of 500, the training time of the research model is 2.3 minutes, the processing time for a single piece of data is 2.3ms, and the resource occupation is 51.3%. Compared with MPC, the training time is shortened by 39.5%, and the resource occupation is reduced by 36.7%. The core of its optimization is that the HLKH module reduces key transmission by 40%. In terms of scalability, when the number of nodes increases from 20 to 100, the model accuracy rate only drops by 6.5%. When 25% of edge nodes fail, the accuracy retention rate of the HAPB-FL model still exceeds 89%, which is superior to the hidden access structure and proxy re-encryption technology proposed by Yin S et al. in 2024 [20]. This performance advantage is attributed to the decentralized architecture of HLKH.

## 6 Conclusion

The HAPB-FL privacy protection model is designed with a hierarchical architecture, covering three core modules: key management, privacy budget allocation, and federated learning training, and collaboratively processes privacy data. Overall, the privacy protection strength, data availability and computational efficiency of the HAPB-FL model are all suitable for the requirements of the IoT. However, through the ablation experiment, it was found that the dynamic noise of APBA led to a 2.9% decrease in accuracy compared with the baseline. The key update of HLKH will increase the communication delay by 25 to 32ms. In terms of scalability, when the number of nodes exceeds 200, the model needs to alleviate the overhead through "subgroup splitting". In the future, we will further explore "subgroup splitting + edge pre-computation" to clarify the universality and expansion boundaries of the research.

## References

[1] P. Muralidhara Rao, and B. D. Deebak. Security and privacy issues in smart cities/industries: technologies, applications, and challenges. Journal of Ambient Intelligence and Humanized Computing, 14(8):10517-10553, 2023. https://doi.org/10.1007/s12652-022-03707-1

[2] Naiyu Wang, Wenti Yang, Xiaodong Wang, Longfei Wu, Zhitao Guan, Xiaojiang Du, and Mohsen Guizani. A blockchain based privacy-

preserving federated learning scheme for Internet of Vehicles. Digital Communications and Networks, 10(1):126-134, 2024. https://doi.org/10.1016/j.dcan.2022.05.020

[3] Jie Wen, Zhixia Zhang, Yang Lan, Zhihua Cui, Jianghui Cai, and Wensheng Zhang. A survey on federated learning: Challenges and applications. International journal of machine learning and cybernetics, 14(2):513-535, 2023. https://doi.org/10.1007/s13042-022-01647-y

[4] Xiao-Kai Cao, Chang-Dong Wang, Jian-Huang Lai, Qiong Huang, and C. L. Philip Chen. Multiparty secure broad learning system for privacy preserving. IEEE Transactions on Cybernetics, 53(10):6636-6648, 2023. https://doi.org/10.1109/TCYB.2023.3235496

[5] Simona Gugliermo, Erik Schaffernicht, Christos Koniaris, and Federico Pecora. Learning behavior trees from planning experts using decision tree and logic factorization. IEEE Robotics and Automation Letters, 8(6):3534-3541, 2023. https://doi.org/10.1109/LRA.2023.3268598

[6] Jie Wen, Zhixia Zhang, Yang Lan, Zhihua Cui, Jianghui Cai, and Wensheng Zhang. A survey on federated learning: challenges and applications. International Journal of Machine Learning and Cybernetics, 14(2):513-535, 2023. https://doi.org/10.1007/s13042-022-01647-y

[7] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. IEEE Communications Surveys & Tutorials, 25(4):2983-3013, 2023. https://doi.org/10.1109/COMST.2023.3315746

[8] Sooraj George Thomas, and Praveen Kumar Myakala. Beyond the cloud: Federated learning and edge AI for the next decade. Journal of Computer and Communications, 13(2):37-50, 2025. https://doi.org/10.4236/jcc.2025.132004

[9] Jingxue Chen, Hang Yan, Zhiyuan Liu, Min Zhang, Hu Xiong, and Shui Yu. When federated learning meets privacy-preserving computation. ACM Computing Surveys, 56(12):32-36, 2024. https://doi.org/10.1145/3679013

[10] Xinyan Li, Huimin Zhao, and Wu Deng. BFOD: Blockchain-based privacy protection and security sharing scheme of flight operation data. IEEE Internet of Things Journal, 11(2):3392-3401, 2023. https://doi.org/10.1109/JIOT.2023.3296460

[11] Yan Huang, Yi Joy Li, and Zhipeng Cai. Security and privacy in metaverse: A comprehensive survey. Big Data Mining and Analytics, 6(2):234-247, 2023. https://doi.org/10.26599/BDMA.2022.9020047

[12] Eva Rodríguez, Beatriz Otero, and Ramon Canal. A survey of machine and deep learning methods for privacy protection in the internet of things. Sensors, 23(3):1252-1275, 2023. https://doi.org/10.3390/s23031252

[13] Badhan Chandra Das, M. Hadi Amini, and Yanzhao Wu. Security and privacy challenges of large language models: A survey. ACM Computing Surveys, 57(6):1-39, 2025. https://doi.org/10.1145/3712001

[14] José Antonio López-Pastor, Miguel Poveda-García, Alejandro Gil-Martínez, David Cañete-Rebenaque, and José Luis Gómez-Tornero. 2-D localization system for mobile IoT devices using a single Wi-Fi access point with a passive frequency-scanned antenna. IEEE Internet of Things Journal, 10(17):14995-15011, 2023. https://doi.org/10.1109/JIOT.2023.3262830

[15] Danny Marks, and John Connell. Unequal and unjust: The political ecology of Bangkok's increasing urban heat island. Urban Studies, 61(15):2887-2907, 2024. https://doi.org/10.1177/00420980221140999

[16] Simone Patonico, An Braeken, and Kris Steenhaut. Identity-based and anonymous key agreement protocol for fog computing resistant in the Canetti-Krawczyk security model. Wireless Networks, 29(3):1017-1029, 2023. https://doi.org/10.1007/s11276-019-02084-6

[17] Alexander Ziller, Tamara T. Mueller, Simon Stieger, Leonhard F. Feiner, Johannes Brandt, Rickmer Braren, Daniel Rueckert, and Georgios Kaissis. Reconciling privacy and accuracy in AI for medical imaging. Nature Machine Intelligence, 6(7):764-774, 2024. https://doi.org/10.1038/s42256-024-00858-y

[18] G. Umarani Srikanth, R. Geetha, and S. Prabhu. An efficient Key Agreement and Authentication Scheme (KAAS) with enhanced security control for IIoT systems. International Journal of Information Technology, 15(3):1221-1230, 2023. https://doi.org/10.1007/s41870-023-01173-2

[19] Panjun Sun, Shigen Shen, Yi Wan, Zongda Wu, Zhaoxi Fang; and Xiao-Zhi Gao. A survey of IoT privacy security: Architecture, technology, challenges, and trends. IEEE Internet of Things Journal, 11(21):34567-34591, 2024. https://doi.org/10.1109/JIOT.2024.3372518

[20] Shoulin Yin, Hang Li, Lin Teng, Asif Ali Laghari, and Vania Vieira Estrela. Attribute-based multiparty searchable encryption model for privacy protection of text data. Multimedia Tools and Applications, 83(15):45881-45902, 2024. https://doi.org/10.1007/s11042-023-16818-4