Wireless Communications and Mobile Computing

Secure Computational Solutions for Sparse Data Challenges in the Internet of Things 2022

Lead Guest Editor: Yan Huang
Guest Editors: Donghyun Kim, Fei Hao, Yan Huo, and Madhuri Siddula

Secure Computational Solutions for Sparse Data Challenges in the Internet of Things 2022

Wireless Communications and Mobile Computing

Secure Computational Solutions for Sparse Data Challenges in the Internet of Things 2022

Lead Guest Editor: Yan Huang
Guest Editors: Donghyun Kim, Fei Hao, Yan Huo, and Madhuri Siddula

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 8545101, 13 pages
https://doi.org/10.1155/2023/8545101

Research Article

Privacy-Preserving Federated Graph Neural Network Learning on Non-IID Graph Data

Kainan Zhang,[1] **Zhipeng Cai**,[1] **and Daehee Seo**[2]

[1]*Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA*
[2]*Department of Computer Science, Sangmyung University, Seoul, Republic of Korea*

Correspondence should be addressed to Zhipeng Cai; zcai@gsu.edu

Received 3 August 2022; Revised 26 September 2022; Accepted 30 September 2022; Published 3 February 2023

Academic Editor: Yan Huo

Since the concept of federated learning (FL) was proposed by Google in 2017, many applications have been combined with FL technology due to its outstanding performance in data integration, computing performance, privacy protection, etc. However, most traditional federated learning-based applications focus on image processing and natural language processing with few achievements in graph neural networks due to the graph's nonindependent identically distributed (IID) nature. Representation learning on graph-structured data generates graph embedding, which helps machines understand graphs effectively. Meanwhile, privacy protection plays a more meaningful role in analyzing graph-structured data such as social networks. Hence, this paper proposes PPFL-GNN, a novel privacy-preserving federated graph neural network framework for node representation learning, which is a pioneer work for graph neural network-based federated learning. In PPFL-GNN, clients utilize a local graph dataset to generate graph embeddings and integrate information from other collaborative clients to utilize federated learning to produce more accurate representation results. More importantly, by integrating embedding alignment techniques in PPFL-GNN, we overcome the obstacles of federated learning on non-IID graph data and can further reduce privacy exposure by sharing preferred information.

1. Introduction

Data providers sometimes share their data to improve the analytical performance of all participants. However, the collaboration among data providers risks privacy leakage of data owners. Insecure data sharing coupled with poor deanonymization is the same as giving away the owner's information for free. Federated learning (FL) is a comparatively different learning strategy that eludes data collection in a centralized location [1], where a typical server model may reveal a user's sensitive data that he/she is not willing to share. Under this concern, FL is aimed at training deep neural networks on multiple local datasets present on local clients without explicitly exposing the data samples to either the central server or cooperating clients.

Graph data is helpful for processing tasks involving complex relationships and dynamic schemata, such as supply chain management and recommendation systems. Although graph neural network stands out by utilizing representation learning to accomplish graph analysis tasks such as node classification and link prediction in the current big data era [2], there are several reasons preventing FL from being widely applied in the domain of graph neural networks. Unlike most earlier federated learning researches with IID computer vision or language data underlying, the non-IID nature of graphs [3] may cause FL resulting in a worse accuracy than the centralized approach when the training dataset becomes large and noisy as real-world graphs and GNNs tend to overfit the training set if it is not properly regularized [4]. Worse, the aggregation mechanism of FL may fail on sparse graphs, where nodes within local neighborhoods provide more noise than useful information for feature aggregation [5]. More broadly, the diversity of the GNNs model makes the current definition of federated GNNs not uniform and unclear [6]. In addition, most of the existing FL algorithms, such as the naive FedAvg algorithm, are designed for the IID dataset, so it is difficult to effectively integrate the information between various clients in common federated

GNN setting [7]. Especially when clients have different sample nodes and cannot share the complete topology information for privacy concerns, applying the leading traditional averaging strategy to the federated process is not suitable because the input nodes of the graph neural networks are different.

Hence, to solve the aforementioned challenges, we propose a novel federated learning framework for graph neural networks with the embedding alignment technique. Because the framework only needs to integrate client-preferred public information, it can significantly reduce the risk of privacy disclosure during the learning process. The embedding alignment technique ensures that the clients holding non-IID data can change information. Furthermore, we find that injecting aligned information into the local model has regularization effects empirically and thus reduces the risk of overfitting. The main contributions of our work are summarized as follows:

(i) We investigate a general training scenario of the federated GNN setting in which multiple clients hold non-IID graph datasets sharing partial structural equivalence

(ii) We propose a novel framework to integrate federated learning and embedding alignment techniques into an end-to-end process flow to obtain accurate embedding results for individual clients

(iii) We conduct extensive experiments on ground truth datasets to prove the effectiveness of the proposed method with the embedding alignment technique and demonstrate the competitive performance of PPFL-GNN framework with respect to noise resistance

2. Related Works

2.1. Federated Learning with Non-IID Dataset. The non-IID local data usually brings statistical challenges for federated learning, which hurts training convergence and significantly reduces accuracy. To conquer the problem, Zhao et al. propose a strategy to improve the training of non-IID data by creating a small portion of data globally shared among all edge devices [8]. To offset the bias introduced by non-IID data and accelerate convergence, Wang et al. propose Favor [9], an experience-driven control framework, which can intelligently select client devices to participate in each round of federated learning. As another research direction, many FL algorithms are proposed to address the problem of learning efficiency under non-IID data settings. FedProx [10] is a generalization and reparametrization of FedAvg, which pioneers in tackling federated network heterogeneity. In FedPD [11], the authors also explore the nonconvex behavior of the FedAvg algorithm and propose a federated learning framework with optimal rates and adaptivity to non-IID data. Similarly, Li et al. propose FedBN [12], which uses local batch normalization to alleviate the feature shift before averaging models with the convergence rate speed-up. However, after conducting extensive experiments, Li et al. [13] find that the current state-of-the-art FL algorithm cannot outperform other algorithms in all cases with comprehensive data partitioning strategies that cover the typical non-IID data cases. Moreover, to achieve differential privacy in federated learning under a non-IID scenario, Xiong et al. design the 2DP-FL algorithm [14] that adds flexible noise to meet various privacy standards. Although these prior methods have been making progress in different fields, none of them consider using the graph with nature non-IID regarding characteristics as the experiment dataset.

2.2. Federated Learning on Graph Neural Networks. Compared with the voluminous progress made in the vision and language domains, researches about federated learning on graphs are still relatively lacking. For example, SGNN [15] uses a similarity-based graph neural network to capture the structural information of nodes, but it only borrows the thought of federated learning to hide the original information from different data sources. More like a variant of federated learning, Lalitha et al. propose a distributed learning algorithm in which nodes update their beliefs by aggregating information from neighbors and learn the most suitable model of the entire network [16]. Recently, the appearance of FedGraphNN [17] promotes federated learning research based on GNN as an open research federated learning system and benchmark. However, their experimental results pose significant challenges in federated GNN training. For example, federated GNNs perform worse in most datasets with a non-IID split than centralized GNNs, indicating that more research is necessary for this field.

Moreover, federated GNN inherits the core problems from traditional federated settings including expensive communication, systems heterogeneity, statistical heterogeneity [18], and privacy concerns [10]. For instance, to handle the statistical heterogeneity of the data, He et al. propose SpreadGNN [19], a novel multitask federated training framework, which can run in the presence of partial labels and no central server by utilizing decentralized periodic averaging SGD to solve decentralized multitask learning problems. Aiming to moderate the privacy concerns, Sajadmanesh and Gatica-Perez develop a privacy-preserving, architecture-agnostic GNN learning algorithm with formal privacy guarantees based on local differential privacy [20], which also aggregates multihop nodes' features to denoise the noisy labels. In addition to general models and theoretical research, the application of federated GNN to practical problems is also worth studying. For example, FedGNN proposes a federated framework for the GNN-based recommendation system [21], which can collectively train GNN models from decentralized user data while using high-level user-item interaction information to preserve privacy. In the remainder of this article, we also address these four challenges in our work and discuss the framework's applicability.

2.3. Embedding Space Alignment. The embedding approach has become a primary topic in machine learning and graphical analysis [22, 23]. Naturally, the alignment of different embedding spaces plays an important role similar to the translation in the communication of different languages. As the pioneer of alignment technique, cross-lingual word

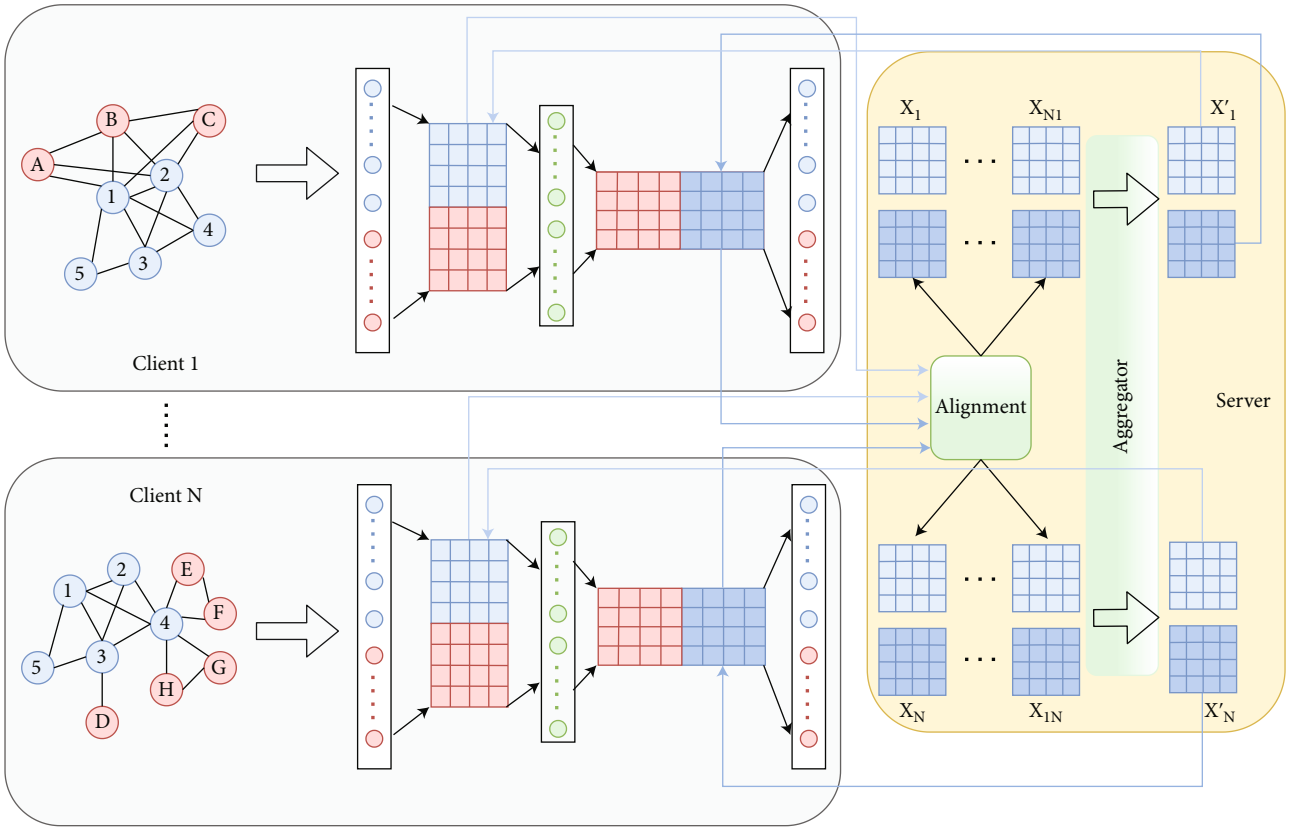


FIGURE 1: Overview of the federated DeepWalk framework. The red nodes are private, and the blue nodes are public. Local training is highlighted in grey, and server aggregation is highlighted in yellow.

embedding alignments have rapidly grown in the past few years [24]. Both MUSE [25] and VecMap [26] provide modern and oft-cited toolkits for bilingual lexical induction (BLI) datasets. With the development of knowledge-driven applications such as question answering and knowledge graph completion, substantial researches on knowledge graph embedding alignments have emerged recently [27, 28]. These thorough studies enlighten us to apply the existing alignment technique, instead of the training target, but as a tool of information extraction and data integration during the training process.

3. Proposed Work

In this section, we first introduce the problem formulation of our work and then explain the details of our approach to learning graph representation in a privacy-preserving way based on two state-of-the-art models.

3.1. Problem Formulation. Denote $\mathbf{C} = \{c_1, c_2, \cdots c_n\}$ as the sets of clients participating in federated learning, and client c_i holds a local undirected graph $G = (\mathbf{U}, \mathbf{E}, \mathbf{F})$ including node set $\mathbf{U}$, edge set $\mathbf{E}$, and node-feature set $\mathbf{F}$. We assume that all the local graphs share a certain amount of nodes defined as public node set $\mathbf{U}_k = \mathbf{U}_1 \cap \mathbf{U}_2 \cap \cdots \cap \mathbf{U}_n$. To protect privacy, each client saves the original data locally, including the edge and attribute information of nonpublic nodes. Only the processed public node information, which is generated as public node embedding by the client's local model, will be uploaded to the server. Our goal is to generate accurate node representation for each client by utilizing federated learning without building and storing the entire graph on the server or client.

3.2. Federated DeepWalk. DeepWalk extends the idea of language modeling to network topology [29], which forms the embryo of graph embedding. Given a random walk sequence composed of network nodes,

$$V_1^n = (v_0, v_1, \cdots, v_n), \tag{1}$$

where $v_i \in \mathbf{U}$. The goal so far is to retrieve the likelihood of observing v_i given the previous $i-1$ nodes in the random walk:

$$\Pr(v_i | (v_1, v_2, \cdots, v_{i-1})). \tag{2}$$

To learn the latent representation, instead of only a probability distribution of node cooccurrence, DeepWalk introduces a mapping function $\Phi : v \in V \longrightarrow R_{|V| \times d}$, which actually is a $|V| \times d$ matrix of free weights serving as the low-dimensional representations of all network nodes in the graph.

Input: $\mathbf{C} = \{c_1, c_2, \cdots c_n\}$: the set of clients
G_i: the local subgraph hold by c_i
$\mathbf{U}_k$: the public nodes shared among $\mathbf{C}$
Output: the matrix of node representation $\Phi_i \in R^{|V| \times d}$ of G_i

LOCAL CLIENTS:
for each client $c_i \in \mathbf{C}$ **do**
  Compute the DeepWalk model weights Φ_i
  Generate the public nodes' embeddings X_i of U_k from Φ_i:
    $X_i = \{\Phi_i(u_1), \cdots, \Phi_i(uk)\}$
  Upload X_i to the server
end for

while not converge **do**
  SERVER:
  for each $i \in k$ **do**
    for each $j \in k (i \neq j)$ **do**
      Align X_j into c_i's space: $X_{ji} = W_{ji} X_j$
    end for
    Aggregate all the aligned embeddings with X_i
      $X_i' = 1/k(\sum_j^k X_{ji} + X_i)$
    distribute X_i' to client c_i for local update
  end for

  LOCAL CLIENTS:
  for each client $c \in \mathbf{C}$ **do**
    Substitute the public nodes' embeddings in Φ_i by X_i'
      $\Phi_i' \longleftarrow (\Phi_i, X_i')$
    Initial the DeepWalk model with Φ_i'
    Compute the model weights Φ_i
  end for
end while
return the matrix of node representation $\Phi_i \in R^{|V| \times d}$ of G*i*

Algorithm 1:The federated DeepWalk framework.

However, the computation is not efficient depending on the length of the random walks. Thus, the SkipGram method in Word2vec [30] is applied to solve the computational problem. Rather than predicting the occurrence of a missing node in the walk, we compute the likelihood of a node appearing as a neighbor in a given window, and the new optimization goal is summarized as follows:

$$\min_{\phi} -\log \Pr(\{(v_{i-w}, \cdots, v_{i-1}, v_{i+1}, \cdots, v_{i+w})\} | \Phi(v_i)), \quad (3)$$

where w is the window size for iterating the possible collocation of the given node v_i. Suppose we deploy DeepWalk as the neural network model in the federated learning setting, then the local client c_i can train a low-dimensional latent representation $R^{|V| \times d}$ of his local graph G_i. After all clients have generated their local graph embeddings, the challenge of a federated learning setting is how all clients collaborate to improve the training results with less disclosure of sensitive information.

In traditional federated learning, each client uploads all weights of the local model to a central server. The central server aggregates these weights to update the global model and then distributes the global model back to the clients. However, in our problem definition, we cannot aggregate all weights directly because each client holds a different subgraph of the global network, which means that the trained latent representations only share commonality on the public nodes partially. Because the potential relationship between public and private nodes is stored in the public nodes' latent representations, as shown in Figure 1, instead of uploading all weights (i.e., the latent representations of all the nodes in the local graph), a client can only upload the weights related to the public nodes (i.e., the latent representations of public nodes), which also carry some sensitive information of the private nodes.

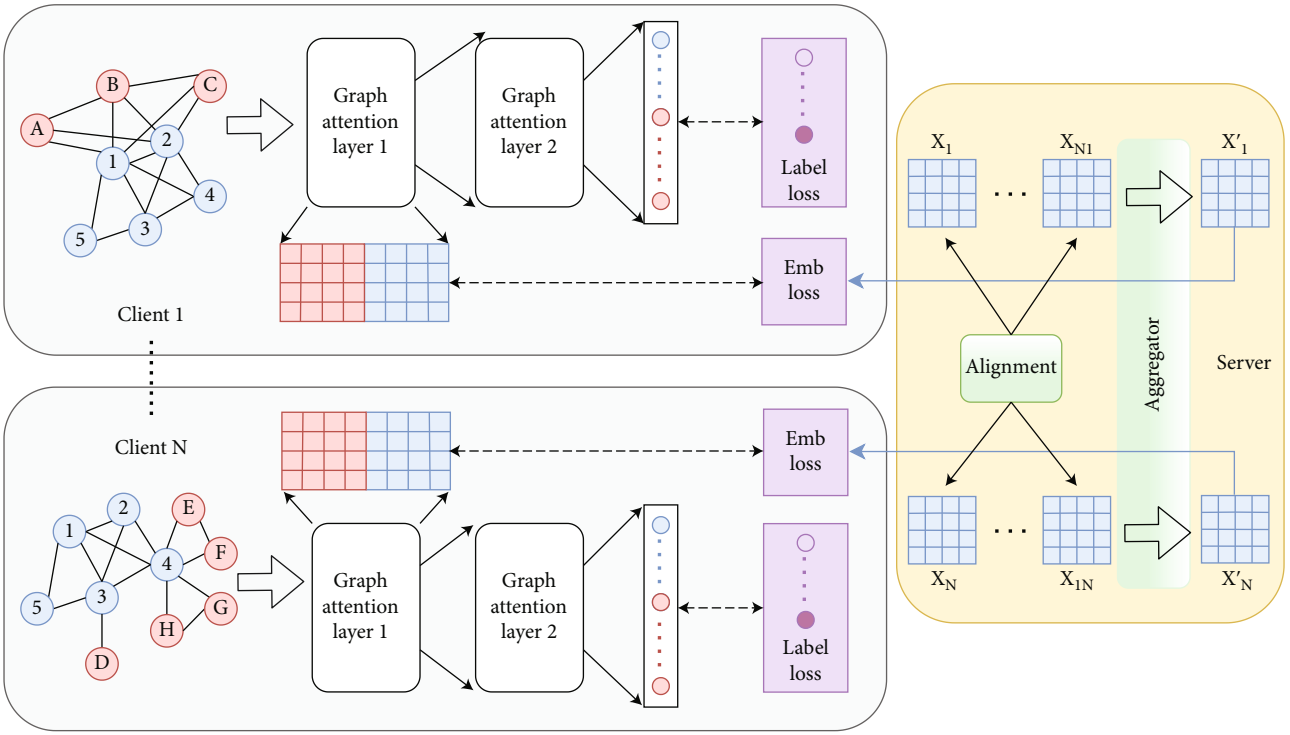


FIGURE 2: Overview of the federated GAT framework. The red nodes are private, and the blue nodes are public. Local training is highlighted in grey, and server aggregation is highlighted in yellow.

Since the latent representations of public nodes are generated from different training graphs, simple aggregation and distribution will break their connections with the unprocessed latent representations of private nodes on the local client. Thus, we apply an embedding alignment technique in the weight aggregation on the central server to convert the latent representations from other clients into a form that the local client understands. For example, there are two local clients c_x and c_y sharing k nodes in the graph. Let $X = \{\Phi_x(u_1), \cdots, \Phi_x(u_k)\}$ and $Y = \{\Phi_y(u_1), \cdots, \Phi_y(u_k)\}, u_k \in \mathbf{U}_k$ be two sets of k public node embeddings coming from c_x and c_y, respectively. For c_y to understand the information of X, we need to align/translate X into the space of c_y, which technically is using a linear mapping matrix W that maps X from the source space c_x to the target space c_y. Furthermore, we can encapsulate the problem to the Procrustes problem [31] and solve it via the singular value decomposition (SVD) of YX^T:

$$W^* = \text{argmin}_{W \in M_d(\mathbb{R})} \|WX - Y\|_F = UV^T, \quad \text{with } U\sum V^T = \text{SVD}(YX^T) \tag{4}$$

where $M_d(\mathrm{R})$ is the $d \times d$ matrix space of real numbers. We denote $X_y = WX$ as the aligned embeddings from source space c_x to target space c_y, and Y_x in the opposite way. The server aggregates X_y and Y to obtain a merged weight Y' and returns Y' to c_y for substituting the current public node embedding vector $\Phi(y_k)$. For multiple clients $\mathbf{C} = \{c_1, c_2, \cdots c_n\}$, the server aligns the embed dings from any pair of clients $\forall c_i, c_j \in \mathbf{C}$ and applies the average aggregation on all the aligned embeddings in the same client's space to get the returning updates for each client. The local clients use the updates as the initial weights to train in a new round. Algorithm 1 summarizes the complete training procedure.

3.3. Federated GAT Framework. GAT [32] introduces an attention mechanism to replace the statically normalized convolution operation in GCN [33]. The input to a single attentional layer is a set of node features, $h = \{\vec{h_1}, \vec{h_2}, \cdots, \vec{h_n}\}, \vec{h_i} \in F$, where n is the number of nodes and F is the node feature set. A linear transformation is firstly applied to every node feature for higher-level expression:

$$z_i^{(l)} = W^{(l)} h_i^{(l)}, \tag{5}$$

where $W^{(l)}$ is a learn-able weight matrix.

Different from the dot product attention mechanism in GCN, GAT applies the additive attention mechanism, which concatenates the z embeddings of two neighbors i and j to compute a pairwise unnormalized attention score $e^{(l)}$ between them. The additive attention mechanism takes the dot product of the concatenation and a weight vector $\vec{a}$ and then applies a LeakyReLU activation function. In order to compare the attention scores with different nodes, a

Input: $\mathbf{C} = \{c_1, c_2, \cdots c_n\}$: the set of clients
G_i: the local subgraph hold by c_i
$\mathbf{U}_k$: the public nodes shared among **C**
Output: the node embeddings H_i' of G_i
1: LOCAL CLIENTS:
2: **for** each client $c_i \in \mathbf{C}$ **do**
3: Compute the GAT model embedding H_i'
4: Generate the public nodes' embeddings X_i of U_k from the intermediate H_i:
5: $X_i = \{H_i(u_1), \cdots, H_i(uk)\}$
6: Upload X_i to the server
7: **end for**
8:
9: **while** not converge **do**
10: SERVER:
11: **for** each $i \in k$ **do**
12: **for** each $j \in k(i \neq j)$ **do**
13: Align X_j into c_i's space: $X_{ji} = W_{ji}X_j$
14: **end for**
15: Aggregate all the aligned embeddings with X_i
16: $X_i' = 1/k(\sum_j^k X_{ji} + X_i)$
17: distribute X_i' to client c_i for local update
18: **end for**
19:
20: LOCAL CLIENTS:
21: **for** each client $c_i \in \mathbf{C}$ **do**
22: Take X_i' as new input weights
23: Compute the GAT-model embedding H_i' with loss L_{new}
24: **end for**
25: **end while**
26: **return** the node embeddings H_i' of G_i

Algorithm 2: The federated GAT framework.

Table 1: The results of the federated DeepWalk framework. LOC indicates the result of local training, FED indicates the result of federated learning, and GLOBAL indicates the cumulative improvement of all clients.

Dataset (classifier)	Client 1		Client 2		Client 3		Client 4		GLOBAL
	LOC	FED	LOC	FED	LOC	FED	LOC	FED	
Cora (MLP)	79.1	80.1	76.0	76.0	74.3	76.1	75.0	76.6	+4.41
Cora (SVC)	79.3	81.4	77.5	78.7	75.5	77.9	75.0	78.0	+8.72
CiteSeer (MLP)	48.1	51.4	46.4	50.4	48.9	51.6	49.2	51.7	+12.5
CiteSeer (SVC)	56.3	60.6	53.5	58.6	57.3	61.6	55.3	60.4	+18.8
Cora (full)	MLP		79.9		SVC		82.0		
CiteSeer (full)	MLP		58.6		SVC		65.4		

normalized coefficient $\alpha^{(l)}$ is computed by the softmax function in the end:

$$\alpha_{ij}^{(l)} = softmax_j\left(e_{ij}^{(l)}\right) = \frac{\exp\left(\text{LeakyReLU}\left(\vec{a}^T\left[z_i^{(l)} \middle\| z_j^{(l)}\right]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\vec{a}^T\left[z_i^{(l)} \middle\| z_k^{(l)}\right]\right)\right)} \quad (6)$$

where N_i is some neighbor of node i in the graph, $\|$ denotes the concatenation operation, and T represents transposition.

Having the normalized attention coefficients calculated, GAT generates the next-level embed-scaled by the attention coefficients.

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}^{(l)} z_j^{(l)}\right) \quad (7)$$

Once we obtain the local embeddings, we have to face the similar challenge of collaborating with different clients in federated learning as the federated DeepWalk framework. Although in DeepWalk model we can extract the

Table 2: The results of the federated GAT framework. Cora_noise is the Cora dataset with noisy labels.

Dataset (classifier)	Client 1		Client 2		Client 3		Client 4		GLOBAL
	LOC	FED	LOC	FED	LOC	FED	LOC	FED	
Cora (MLP)	84.4	86.3	85.7	85.8	83.1	83.8	85.7	86.0	+3.01
Cora (SVC)	86.1	86.7	86.5	86.0	85.6	85.0	85.4	86.0	+0.1
Cora_noise (MLP)	81.4	81.0	79.0	79.5	72.2	75.9	70.6	77.3	+10.4
Cora_noise (SVC)	82.0	82.4	80.0	80.8	74.5	77.4	72.3	78.3	+10.1
CiteSeer (MLP)	72.3	74.7	72.8	74.1	72.8	74.3	72.7	75.0	+7.51
CiteSeer (SVC)	72.8	74.5	72.3	74.3	72.1	74.3	72.1	74.1	+7.82
Cora (full)	MLP		86.2		SVC		86.1		
CiteSeer (full)	MLP		74.7		SVC		74.8		

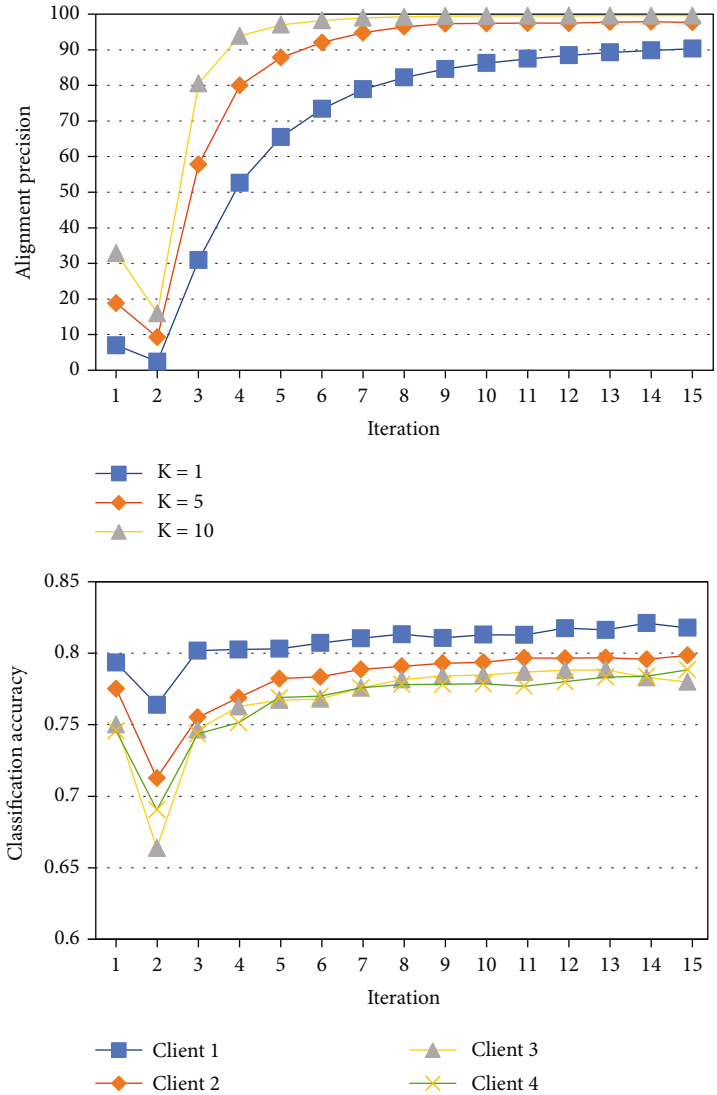


Figure 3: The KNN alignment precision and SVC classification accuracy corresponding to each iteration of the federated DeepWalk framework on the Cora dataset.

public nodes' embeddings from the model weights directly, the weights of GAT integrate both public and private information and cannot be split directly by node's category. Therefore, as shown in Figure 2, we upload the public nodes' embeddings X coming from the model's intermediate layer (i.e., $h_i^{(l)}, i \in \mathbf{U}_k$) to the server without exposing the model weights. Then, the server executes the same processes in the federated DeepWalk framework to align, aggregate, and

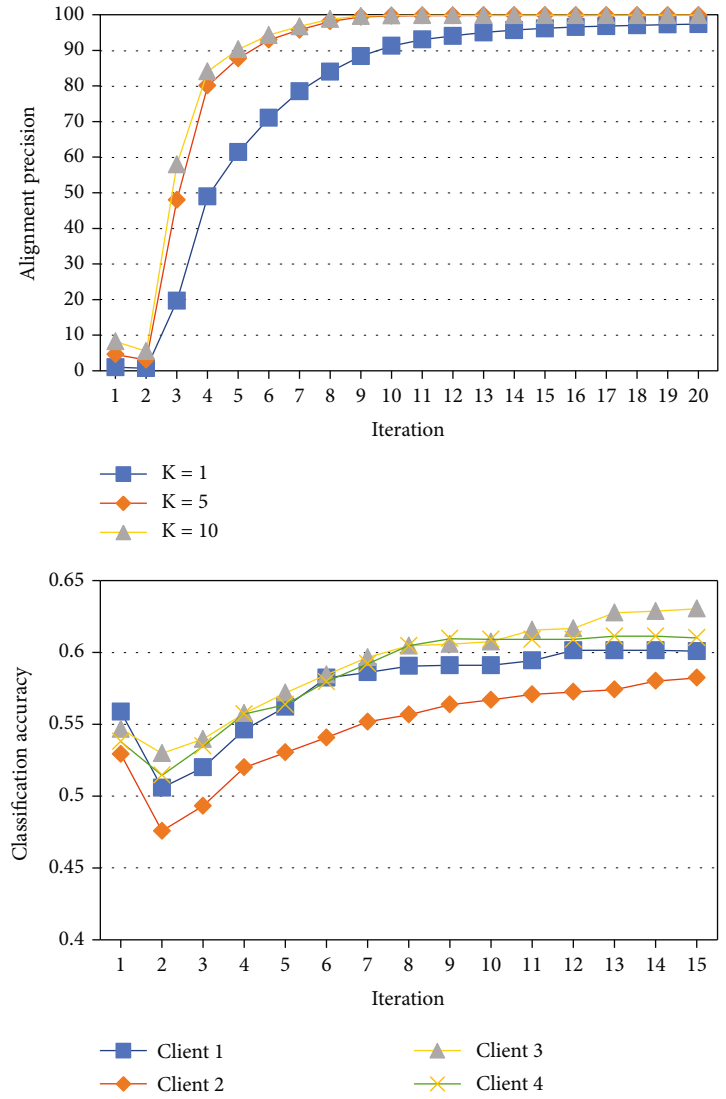


Figure 4: The KNN alignment precision and SVC classification accuracy corresponding to each iteration of the federated DeepWalk framework on the CiteSeer dataset.

distribute the updates X' to clients. As we cannot use the aligned embedding to manipulate GAT's model weights directly, another cosine-embedding loss L_{emb} is added beside the original cross-entropy loss L_{label} to integrate the information of X' back into the model.

$$L_{emb} = 1 - \cos\left(X, X'\right)$$
$$L_{label} = -\frac{1}{N}\sum_{i=1}^{n} y_i \log\left(\hat{y}_i\right) \tag{8}$$

where y_i is the one-hot label of each node and $\hat{y}_i = \text{soft max}\left(h_i^{(l+1)}\right)$. Thus, the new loss of local training is the combination of the cosine-embedding loss and the cross-entropy loss:

$$L_{\text{new}} = L_{\text{label}} + \beta L_{\text{emb}}, \tag{9}$$

where β is a model hyperparameter to balance the local information preservation and the external information integration. During the experiment, we observe that the last attention layer is so powerful that it overwhelms the cosine-embedding loss of the final output $h^{(l+1)}$. Hence, based on two-stage CNN training [34] and federated split learning [35], we inject the external information via the intermediate layer $h^{(l)}$ at an earlier stage. Algorithm 2 summarizes the complete procedure of the federated GAT framework.

4. Experiments

In this section, we present the experiments developed by PyTorch and conducted on a workstation with an Intel Core i7 2.80 GHz CPU and a NVIDIA GeForce GTX 1070 GPU.

4.1. Datasets. In our experiments, we employ two datasets, Cora [36] and CiteSeer [37], which are commonly used in the GNN research. The Cora dataset includes 2,708

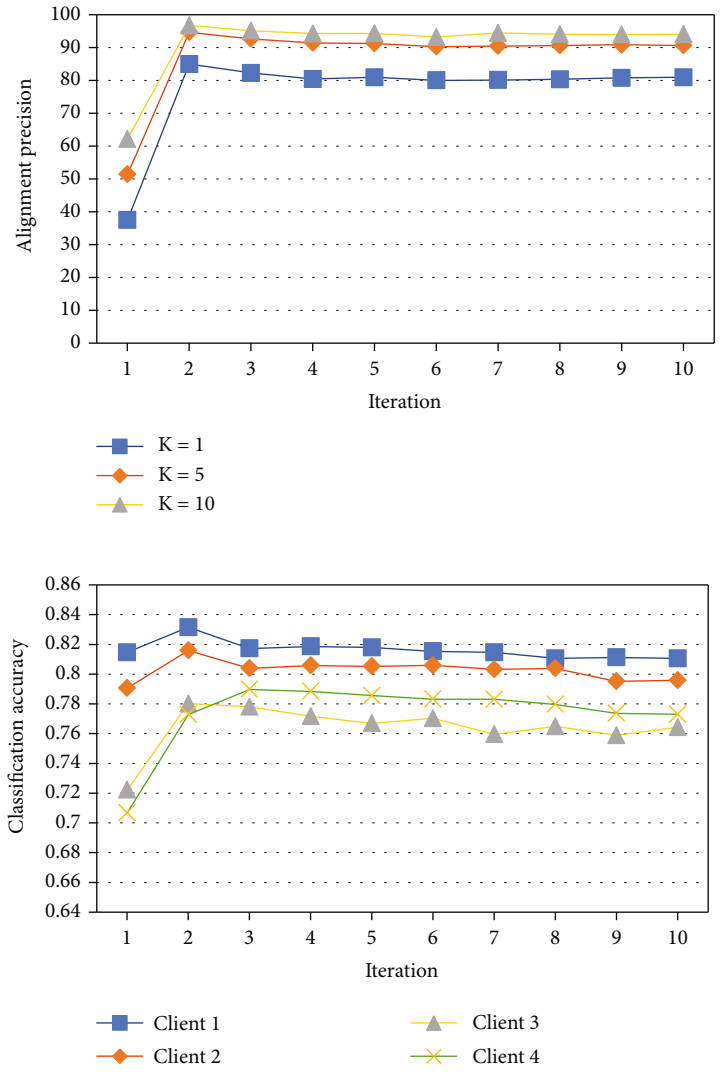


Figure 5: The KNN alignment precision and SVC classification accuracy corresponding to each iteration of the federated GAT framework on the Cora_noise dataset.

publications as nodes classified into seven classes, and its citation network consists of 5,429 links. The CiteSeer dataset includes 3,327 nodes classified into six classes, and its citation network consists of 4,732 links. Both datasets use unique words in each document as the node features. We set up four clients participating in the federated learning, so each dataset is split into four subgraphs with an equal number of nodes and assigned to each client. By default, each Cora subgraph has 1,489 (55%) nodes in total with 1,083 (40%) public nodes and 406 (15%) private nodes, while each CiteSeer subgraph has 1,829 (55%) nodes in total with 1,330 (40%) public nodes and 499 (15%) private nodes. (%) shows the percentage of the nodes in the original graph.

4.2. Baselines and Metrics. We use the client's local training as the baseline to verify whether our framework can improve each client's graph embedding result through collaborative training. In the DeepWalk-based training, all clients use the same model architecture with randomly initialized weights for local training or federated training, while in the GAT-based training, clients use the original GAT model for the local training and the modified GAT model with embedding loss for the federated training. Other architecture and parameters are fixed in a controlled experiment.

For all the implementations, we embed each graph into a 16-dimensional space and run the experiments on the classification tasks to evaluate the quality of the embedding by applying one multilayer perceptron (MLP) classifier and another support vector classifier (SVC) implemented in the Python module scikit-learn to predict the label of a node. For all the experiments, we use 5-fold cross-validation to ensure models' reliability and effectiveness, and the classification results of the two frameworks are given as micro F1-scores.

4.3. Performance Evaluation. The federated DeepWalk framework results are presented in Table 1. We can observe that in contrast to the embeddings generated by limited local information, both classifiers can achieve higher classification

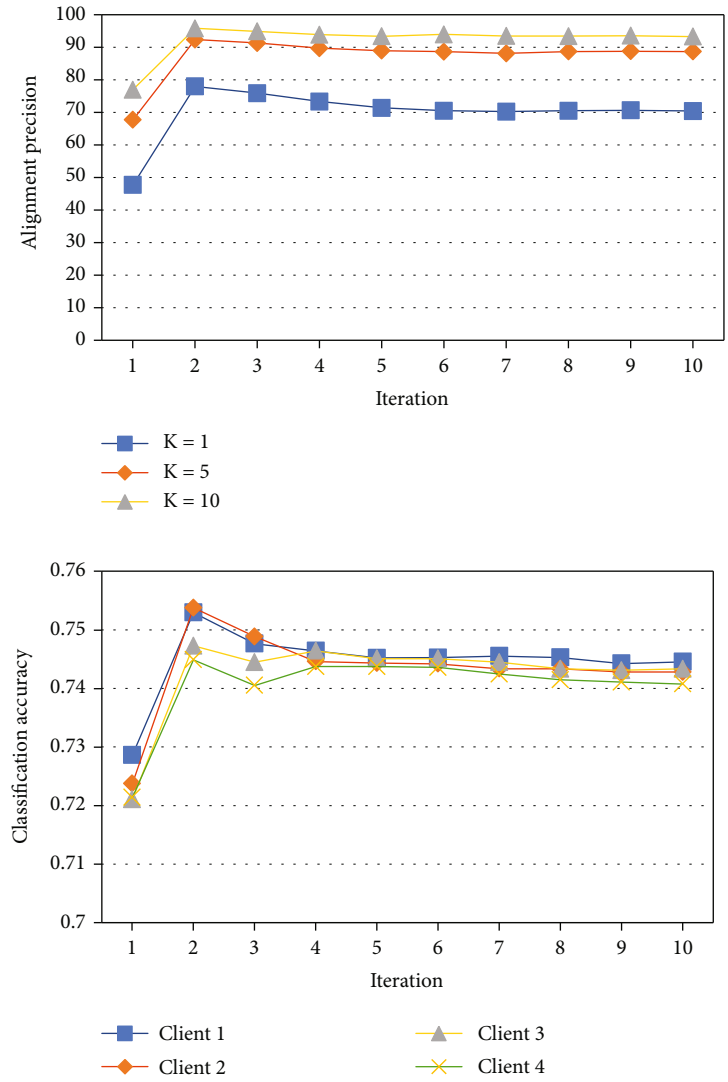


Figure 6: The KNN alignment precision and SVC classification accuracy corresponding to each iteration of the federated GAT framework on the CiteSeer dataset.

accuracy on the embeddings trained by full use of the graph data. Under this precondition, the proposed FL methodology can improve the global results by 4.41% (MLP) and 8.72% (SVC) on the Cora dataset, while 12.5% (MLP) and 18.8% (SVC) on the CiteSeer dataset, respectively. Specifically, every client in the CiteSeer experiment receives steady improvement compared with the local baseline.

For the federated GAT framework, we set the hyperparameter $\beta = 1$ for the Cora dataset and $\beta = 0.75$ for the CiteSeer dataset. As shown in Table 2, we obtain similar results on the CiteSeer dataset with 7.51% (MLP) and 7.82% (SVC) accuracy improvements. Because GAT uses weighting neighbor features with feature-dependent and structure-free normalization, which does not rely on knowing the entire graph structure in advance, the local client can generate favorable embedding by partial information of a denser Cora dataset. Thus, our method is subject to further refining the embedding in this case. In addition to cleaning the Cora dataset, we randomly modify 15% labels as noisy (incorrect) labels during the training, leading to a considerable performance loss for clients such as client 3 and client 4. However, our proposed method can effectively mitigate the influence of noisy labels by integrating information from other clients. Consequently, the poor performance of client 3 or client 4 receives a significant improvement.

4.4. Impact of Alignment on Performance. To demonstrate the effectiveness of applying alignment during the FL aggregation, we plot the alignment precision of the public latent representations and the SVC classification accuracy of the graph embeddings corresponding to each training iteration of both frameworks in Figures 3–6. We use k-nearest neighbors with $k = 1$, 5, and 10 to measure the alignment precision between any pair of the public latent representations. Because we need to align each local representation to the dimension of the other clients, there are 12 pairs in the four clients' settings, and we only show the average value of 12 alignments in the figures as the variance is slight.

Table 3: Results of the different shared public nodes.

Percent	DeepWalk			GAT		
	LOC	FED	Diff	LOC	FED	Diff
5%	57.1	61.8	+4.7	74.4	75.1	+0.7
10%	57.3	62.3	+5.0	73.9	75.1	+1.2
20%	61.8	65.0	+3.2	71.8	73.5	+1.9
30%	63.4	65.8	+2.4	70.5	72.6	+2.1
40%	67.8	69.2	+1.4	70.5	73.0	+2.5

For the federated DeepWalk framework in Figures 3 and 4, although the classification accuracy of locally trained graph embedding is acceptable in the initial iteration, their alignment results are inferior because of the random initialization. Consequently, we cannot integrate the information of different clients effectively, which leads to the performance diving in the second iteration. However, the rough integration in the first two iterations helps in the united initialization by setting the tone for the subsequent training. Thus, we observe that the quality of graph embedding improves with the promotion of the alignment effect, which can achieve above 90% precision of $k = 1$ at the convergence stage. For the federated GAT framework in Figures 5 and 6, the initial representation alignment results are satisfactory with a fair classification accuracy of graph embedding. Moreover, we observe both alignment precision and classification accuracy surge in the second iteration after the federated learning process. Nevertheless, as we only use cosine-embedding loss at the intermediate layer, partial integrated information is squeezed out when the federated procedure converges within ten iterations. In general, there is a positive correlation between the alignment precision and classification accuracy, which confirms the effectiveness of our method.

5. Discussion

Through previous experiments, we find that our method performs better when applied to the CiteSeer dataset, which is more sparse than the Cora dataset relatively. Denser subgraphs mean the local clients have more information, limiting the improvement effect of federated learning. However, if the degrees of the shared nodes are low, they cannot comprehensively transmit the local information during the integration. Therefore, we design the supplemental experiments to further study the suitable application scenarios. Instead of randomly generating the subgraphs and selecting 40% public nodes to share, we compose the subgraphs with different percentages of top high-degree nodes from the original graph as the public nodes. We conduct the same embedding classification experiment and render the average accuracy of four clients in Table 3.

Under the DeepWalk framework, the classification accuracy of locally generated graph embeddings increases as the degree of nodes in the subgraph increases. Although federated learning can still improve the overall classification effect, the magnitude of improvement diminishes. With a simpler model DeepWalk, the local clients are more likely to get an underfit model with inferior prediction accuracy below 70%. Federated learning tackles the underfitting issue more by sharing public information between clients and indirectly increasing the local training dataset's size. In the experimental group of GAT, we notice that the higher subgraph density reduces the accuracy of local graph embedding. One reason is that the subgraphs generated by our method are disassortative, and the local aggregation mechanism of GAT may fail on disassortative graphs, where nodes within local neighborhoods provide more noise than helpful feature information. Another reason is that the local model is overfitting the denser training subgraph. However, the federated learning setting prevents the local model from focusing on the training data, and the embedding alignment technique has regularization effects empirically to avoid overfitting. Overall, our approach is suitable for general application scenarios, and the improvement effect is more prominent when the local embedding effect is unsatisfied.

6. Conclusions

This paper investigates a practical problem of federated graph neural networks with non-IID datasets and proposes a novel federated learning framework. Through embedding alignment, we can normalize the common latent representation of each client as uniformly as possible and enable information integration in a federated setting. The experimental results demonstrate that our framework can achieve higher data usability than local training with privacy preservation. Other advanced embedding alignment technologies can be explored in future work for more accurate information integration. Investigation of the shared public nodes is still worthwhile. For future expansion, discovering an optional composition of public nodes to reduce the number of shares can better balance privacy protection and data availability.

Data Availability

The Cora dataset consists of 2,708 scientific publications classified into one of seven classes. The citation network consists of 5,429 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1,433 unique words. (original source: http://web.archive.org). The CiteSeer dataset consists of 3,312 scientific publications classified into one of six classes. The citation network consists of 4,732 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 3,703 unique words (introduced by C. Lee Giles et al. in CiteSeer: An Automatic Citation Indexing System).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean Government under grant 20ZR1300 (Core Technology Research on Trust Data Connectome). This work was also supported by the National Science Foundation (No. 1704287).

References

[1] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: strategies for improving communication efficiency," 2016, https://arxiv.org/abs/1610.05492.

[2] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[3] W. Zhang, J. C. Weiss, S. Zhou, and T. Walsh, "Fairness amidst non-IID graph data: a literature review," 2022, https://arxiv.org/abs/2202.07170.

[4] K. Zhou, Y. Dong, W. Lee, B. Hooi, H. Xu, and J. Feng, "Effective training strategies for deep graph neural networks," 2020, https://arxiv.org/abs/2006.07107.

[5] Y. Ye and S. Ji, "Sparse graph attention networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, pp. 905–916, 2021.

[6] P. Kairouz, H. B. McMahan, B. Avent et al., "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1-2, pp. 1–210, 2021.

[7] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," 2019, https://arxiv.org/abs/1907.02189.

[8] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, https://arxiv.org/abs/1806.00582.

[9] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-IID data with reinforcement learning," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pp. 1698–1707, Toronto, ON, Canada, 2020.

[10] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[11] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, "FedPD: a federated learning framework with optimal rates and adaptivity to non-IID data," 2020, https://arxiv.org/abs/2005.11418.

[12] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "FedBN: federated learning on non-IID features via local batch normalization," 2021, https://arxiv.org/abs/2102.07623.

[13] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-IID data silos: an experimental study," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 965–978, Kuala Lumpur, Malaysia, 2022.

[14] Z. Xiong, Z. Cai, D. Takabi, and W. Li, "Privacy threat and defense for federated learning with non-i.i.d. data in AIoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, 2022.

[15] G. Mei, Z. Guo, S. Liu, and L. Pan, "SGNN: a graph neural network based federated learning approach by hiding structure," in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 2560–2568, Los Angeles, CA, USA, 2019.

[16] A. Lalitha, O. C. Kilinc, T. Javidi, and F. Koushanfar, "Peer-to-peer federated learning on graphs," 2019, https://arxiv.org/abs/1901.11173.

[17] C. He, K. Balasubramanian, E. Ceyani et al., "FedGraphNN: a federated learning system and benchmark for graph neural networks," 2021, https://arxiv.org/abs/2104.07145.

[18] J. Pang, Y. Huang, Z. Xie, Q. Han, and Z. Cai, "Realizing the heterogeneity: a self-organized federated learning framework for IoT," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3088–3098, 2021.

[19] C. He, E. Ceyani, K. Balasubramanian, M. Annavaram, and S. Avestimehr, "SpreadGNN: serverless multi-task federated learning for graph neural networks," 2021, https://arxiv.org/abs/2106.02743.

[20] S. Sajadmanesh and D. Gatica-Perez, "Locally private graph neural networks," in *CCS '21: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2130–2145, Seoul, South Korea, 2021.

[21] C. Wu, F. Wu, Y. Cao, Y. Huang, and X. Xie, "FedGNN: federated graph neural network for privacy-preserving recommendation," 2021, https://arxiv.org/abs/2102.04925.

[22] K. Li, G. Lu, G. Luo, and Z. Cai, "Seed-Free Graph De-Anonymiztiation with Adversarial Learning," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20)*, pp. 745–754, New York, NY, USA, 2020.

[23] K. Li, G. Lu, G. Luo, and Z. Cai, "Adversarial privacy-preserving graph embedding against inference attack," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 8, pp. 6904–6915, 2021.

[24] S. Ruder, I. Vulic, and A. Søgaard, "A survey of cross-lingual word embedding models," *Journal of Artificial Intelligence Research*, vol. 65, pp. 569–631, 2019.

[25] A. Conneau, G. Lample, M.'. A. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," 2018, https://arxiv.org/abs/1710.04087.

[26] G. Dinu, A. Lazaridou, and M. Baroni, "Improvng zero-shot learning by mitigating the hubness problem," 2014, https://arxiv.org/abs/1412.6568.

[27] Z. Sun, W. Hu, Q. Zhang, and Y. Qu, "Bootstrapping entity alignment with knowledge graph embedding," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, pp. 4396–4402, Stockholm, Sweden, 2018.

[28] Q. Zhang, Z. Sun, W. Hu, M. Chen, L. Guo, and Y. Qu, "Multi-view knowledge graph embedding for entity alignment," 2019, https://arxiv.org/abs/1906.02390.

[29] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, New York, USA, 2014.

[30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[31] C. Wang and S. Mahadevan, "Manifold alignment using Procrustes analysis," in *Proceedings of the 25th international conference on Machine learning - ICML '08*, pp. 1120–1127, Madison, Wisconsin, USA, 2008.

[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph Attention Networks," in *International Conference on Learning Representations*, Vancouver, BC, Canada, February 2018.

[33] M. Welling and T. N. Kipf, "Semisupervised classification with graph convolutional networks," in *J. International Conference on Learning Representations (ICLR 2017)*, Toulon, France, 2016.

[34] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan, "Cascade residual learning: a two-stage convolutional neural network for stereo matching," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 887–895, Venice, Italy, 2017.

[35] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, "Splitfed: when federated learning meets split learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 8485–8493, 2022.

[36] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. EliassiRad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[37] C. L. Giles, K. D. Bollacker, and S. Lawrence, "CiteSeer: an automatic citation indexing system," in *Proceedings of the third ACM conference on Digital libraries - DL '98*, pp. 89–98, Pittsburgh, Pennsylvania, USA, 1998.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2659901, 12 pages
https://doi.org/10.1155/2022/2659901

Research Article

Lifetime-Maximized Strong Barrier Coverage of 3D Camera Sensor Networks

Yi Hong[1,2] **Chuanwen Luo**[1,2] **Deying Li**[3] **Zhibo Chen**[1,2] **and Xiyun Wang**[1]

[1]*School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China*
[2]*Engineering Research Center for Forestry-Oriented Intelligent Information Processing of National Forestry and Grassland Administration, Beijing 100083, China*
[3]*School of Information, Renmin University of China, Beijing 100872, China*

Correspondence should be addressed to Chuanwen Luo; chuanwenluo@bjfu.edu.cn

Received 26 July 2022; Accepted 6 September 2022; Published 19 September 2022

Academic Editor: Yan Huang

Camera sensor networks (CSNs) have advantages on providing the precise and multimedia information for plenty of applications. The high coverage quality of CSNs especially satisfies the monitoring requirements of barrier coverage. In three-dimensional (3D) application scenarios, the tracking of the potential intruder in the monitored irregular spaces brings more difficulties and challenges on strong barrier coverage for CSNs. In this paper, we consider the strong barrier coverage problem in 3D CSNs and focus on the objective of monitoring the intruder with high resolution and maximizing the network lifetime. We firstly introduce the definition and hardness proof for the problem based on the irregular space model and the network model, which adopts the Region of Interest (ROI) sensing model with high effective resolution. Secondly, we design two sleep-and-awake scheduling algorithms for the problem in homogeneous and heterogeneous networks, respectively, which are based on the auxiliary graph transformation and the disjoint flows construction. To evaluate these algorithms' performance on the lifetime maximization, we conduct extensive simulation experiments and analyze their results on their advantages and applicable scenarios.

1. Introduction

Wireless sensor networks (WSNs) are being studied for a long time, which can be classified into sensor-based studies and data-based works. The most data-based works focused on the extracting kernel dataset and data query processing [1, 2]. The most sensor-based studies concentrated on the coverage and data transmission issues. With the high accuracy of monitoring information on coverage issue, camera sensor networks (CSNs) have been utilized for a wide range of applications which can be classified into indoor monitoring [3] and outdoor surveillance, e.g. military inspection and wild animal protection. For the military applications, CSNs can provide intrusion warning and action trend prediction for constructing the military boundaries to guarantee the quality of coverage service. For the wild animal protection, CSNs do not only prevent illegal personnel from entering the protection regions for illegal poaching but also avoid the protected animals from escaping from the regions. Thus barrier coverage has got a lot of attentions for research. To satisfy the coverage requirement of the practical applications, barrier coverage has the highest requirement on the sensed information in these coverage optimization problems.

Among the related theoretical research of barrier coverage scheduling in CSNs, the most concerned issues are the particularity of the monitored space and the accuracy of the sensing model. For the particularity of the monitored space, the space may have an irregular terrain structure in the applications like mountainous regions as shown in the orange boundary in Figure 1. The complex structure brings the difficulties and challenges for the sensor deployment and the sensing model construction, which should be considered in the sensor scheduling. For the accuracy of the sensing model, the most existing works adopted the full-view sensing model proposed in

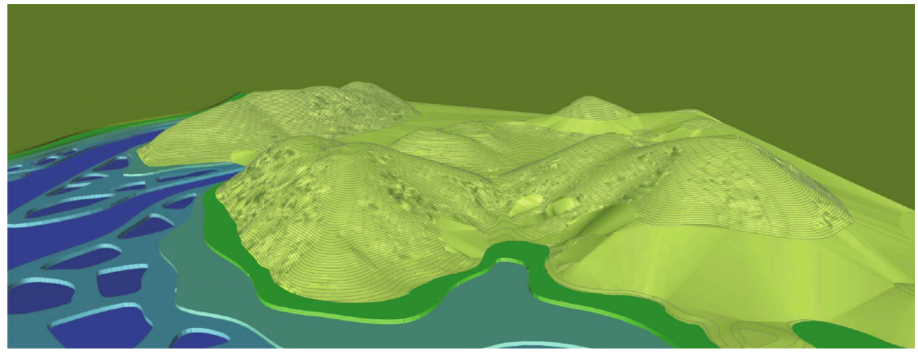


Figure 1: An instance of barrier coverage in forest region scenarios.

[4], which can guarantee the coverage for the target's all facing directions. Besides the sensing omnidirectivity, high image resolution is also considered in the sensing model in [5].

In this paper, we study on the barrier coverage problem in 3D CSNs for the application with irregular geometrical characteristics and strong coverage requirements. The goal of the problem is maximizing the network lifetime under the premise in strong barrier coverage, called the Lifetime-Maximized Strong Barrier Coverage problem for 3D CSNs (LifMax-BC Problem). To solve the problem, the modelling of the irregular space is our first consideration. And we secondly consider the sensing model of the camera sensors in [5] which is being modeled based on the combination of the sensing region and the image resolution. Thirdly, we focus on the problem in different network conditions, homogeneous networks and heterogeneous networks. The list of our contributions is as follows.

(i) We introduce LifMax-BC Problem for strong barrier coverage in 3D CSNs based on modelling the monitored space and the sensing region and give its hardness proof;

(ii) We propose two scheduling algorithms with the sleep-and-awake mode to solve the problem in the homogeneous networks and the heterogeneous networks, which are based on the auxiliary transformation and the disjoint maximum flow construction;

(iii) We conduct a large number of experiments and evaluate the performance of the proposed algorithms in terms of the constructed barrier number. Based on the simulation results, we analyze each scheduling algorithm's applicable scenarios.

The rest of the paper is organized as follows: Section 2 presents the related works. Section 3 introduces the preliminaries, the definition of our problem and the NP-hardness proof. The two scheduling algorithms are, respectively, proposed in Section 4 and Section 5. Performance evaluations are given in Section 6. concludes this paper and discusses the future work.

2. Related Works

The existing sensor-based research on wireless sensor networks (WSNs) can be classified into coverage problems and data transmission problems [6, 7], in which coverage includes target coverage, area coverage, and barrier coverage. Many mature sensing models of camera sensors have been formed from the contribution of the works on target and area coverage in WSNs.

Based on the sensing model of different kinds of camera sensors, there are more studies on barrier coverage with different optimization goals. Based on the sensing model of directional sensors, Wang and Cao [8] studied the construction problem for strong barrier coverage and presented redundancy reduction techniques. The authors proposed the algorithms to solve the barrier coverage problems with the minimum coverage cost based on modeling the full-view-covered regions in [9]. And Mohammad et al. [10] proposed a centralized barrier constructing algorithms based on distributed learning automata for adjustable-orientation directional sensor networks.

Among the optimization goals of barrier coverage, lifetime maximization and robustness guarantee have got attentions beside the coverage cost minimization. For lifetime maximization, Zhang et al. [11] designed a scheduling algorithm for maximizing the full-view coverage duration to solve the fairness-oriented coverage maximization problem, which is based on the full-view sensing model. For robustness guarantee, there exists lots of works to build k-barrier coverage. The k-barrier coverage algorithm for one-dimensional scenarios was proposed in [12], and the k-barrier coverage algorithm for one-dimensional scenarios was designed in [13]. For the sensors with the movement constraints, the strategy was presented for the maximum k -barrier coverage problem in [14]. There are some solutions for barrier coverage problem based on the classical theories: based on the divide and conquer theory, Wen et al. [15] proposed an efficient algorithm to construct k-barrier; based on the Dijkstra algorithm, Liu et al. [16] realized the minimum full-view coverage for mobile CSNs. And there are some algorithms for meeting special requirements in barrier coverage problem: the sensor interference issue was solved in the algorithm for k-barrier coverage in [17]; the one-way k -barrier algorithm was proposed in [18] to avoid one-way

invasion; the strategy for filling barrier coverage holes was designed to realize the goal of minimizing the energy consumption [19].

Considering the robustness guarantee, we focus on the strong barrier coverage problem with the goal of lifetime maximization and we will design two heuristics for CSNs with homogeneous networks and heterogeneous ones.

3. Preliminaries and Problem Formulations

3.1. Space Model. For the applications of CSNs in 3D scenarios, the monitored space can be modeled as a regular cube or cuboid in the most recent related works. The regular model of the space is beneficial to the deployment of the camera sensors and the construction of the sensing model of sensors. However, for the most applications with the rugged terrain or the mountain topography, the spaces are much different from those with the flat topography. The regular model of the space cannot provide precise position for the deployment of the sensors, which will affect the evaluation of the coverage quality.

With the consideration of the topographic complexity of the monitored space in real scenarios, we model the monitored space into an irregular 3D curve strip $\mathcal{ST}$ instead of a regular cube or cuboid. The 3D curve strip space $\mathcal{ST}$ has two terminal sections $\mathcal{S}$, $\mathcal{D}$ and the ceiling and the ground planes $\mathcal{T}$, $\mathcal{B}$, which can be indicated as a quadruple $\mathcal{ST} = (\mathcal{S}, \mathcal{D}, \mathcal{T}, \mathcal{B})$ as shown in Figure 2. Note that if the irregular 3D strip is a cyclic annular or a zigzag band, it can be decomposed into multiple curve strips which are similarly modeled in the paper.

3.2. Sensing Model Based ROI and Network Model. For the sensing model of camera sensors, the full-view coverage model has been widely applied for the most two-dimensional scenarios, which was introduced in [4]. The full-view coverage model can provide the omnidirectional coverage based on the facing directions of the targets, which can guarantee high coverage accuracy.

In this paper, considering the accuracy of the capturing information and 3D application scenarios, we adopt the Region of Interest (ROI) sensing model with high effective resolution, which was proposed in the research [5] as the 3D sensing model of camera sensors. The ROI sensing model of camera sensors dose not only consider the monitored target's facing direction and position height but also construction the relationship between the 3D coverage space and the 2D projection area for the camera sensor. The model can satisfy the coverage requirement of application and the strategy design of sensor scheduling.

The ROI sensing model is applied in a 3D curve strip space $\mathcal{ST}$ in our paper. Considering a pair of a camera sensor v and a target t in $\mathcal{ST}$, we focus on three groups of parameters: (1) the heights of v and t, denoted as H and h, respectively; (2) the length of target t, denoted as L, which is decide by the target itself; and (3)the angle between the t's facing direction and the v's viewing direction in the vertical plane (Effective vertical angle), denoted by β. To guarantee the effective coverage, the parameter has a range with the minimum effective vertical angle $\beta_{\min}$ and the maximum effective vertical angle $\beta_{\max}$, which is determined by the required resolution and can be predefined. Based on the conclusion in [5], the angle between the target's facing direction and the sensor's viewing direction in the horizontal plane is out of the consideration because of the instability of its value. The definition of ROI model for the camera sensor is given as follows:

Definition 1 (The ROI Sensing Model of Camera Sensors). Consider a 3D curve strip space $\mathcal{ST} = (\mathcal{S}, \mathcal{D}, \mathcal{T}, \mathcal{B})$, a camera sensor v located at $(\mathrm{X}, \mathrm{Y}, \mathrm{H})$ and a target t located at $(\mathrm{x}, \mathrm{y}, \mathrm{h})$ in $\mathcal{ST}$, the effective projection sensing area of v for t is a sector-shaped ring or an annular-sector domain $\mathrm{SecRing} = (\mathrm{X}, \mathrm{Y}, \mathrm{r}, \mathrm{R})$ on $\mathcal{B}$ with the inside radius $\mathrm{r} = H - h + (L/2)/\tan\beta_{\max}$ and the external radius $\mathrm{R} = H - h + (L/2)/\tan\beta_{\min}$ as shown in Figure 3.

Based on the ROI sensing model of the camera sensor, we consider the target or intruder in the barrier coverage with the known height and length, i.e., h and L. For example, if the monitored intruder is a person, the person's height and face length can be set as 1.7 meters and 0.5 meters, respectively.

The camera sensor network considered in our paper is composed of N randomly-deployed nodes, which are candidate for barrier coverage scheduling for a 3D curve strip space $\mathcal{ST} = (\mathcal{S}, \mathcal{D}, \mathcal{T}, \mathcal{B})$. These N nodes are collected in the set V. For each node v_i in V, it has its own position (X_i, Y_i, H_i) and the maximum working duration l_i. With the consideration of the complexity and irregularity of the monitored space, the heights of the sensors are different. And we will discuss the cases with the same working duration and the different working durations later.

For the network, the camera sensors can be modeled as a node set $V = \{v_1, v_2, \cdots, v_N\}$. And we consider the connectivity between each pair of nodes based on their sensing ranges. Based on the ROI sensing model of nodes, there is an edge $e_{ij} = (v_i, v_j)$ between v_i and v_j if their sector-shaped rings intersect, i.e., $\mathrm{SecRing}_i \cap \mathrm{SecRing}_j \neq \varnothing$. All the connected edges are collected into the edge set E. Then the original network is modeled as $G = (V, E)$ as shown in Figure 4. And it is assumed that the transmission radius is at least twice of the sensing radius for each camera sensor, then the network $G = (V, E)$ is a connected graph.

3.3. Problem Definitions and Hardness. We focus on the strong barrier coverage in 3D CSNs, which can guarantee to detect intruders without any constraint on crossing paths in the boundary space. Based on the preliminaries, we propose the Lifetime-Maximized Strong Barrier Coverage problem for 3D CSNs (LifMax-BC Problem), whose formal definition is as follows.

Definition 2 (LifMax-BC Problem) Given.

(i) A 3D continuous curve strip space $\mathcal{ST} = (\mathcal{S}, \mathcal{D}, \mathcal{T}, \mathcal{B})$ where $\mathcal{S}$, and $\mathcal{D}$ are $\mathcal{ST}$'s two terminal

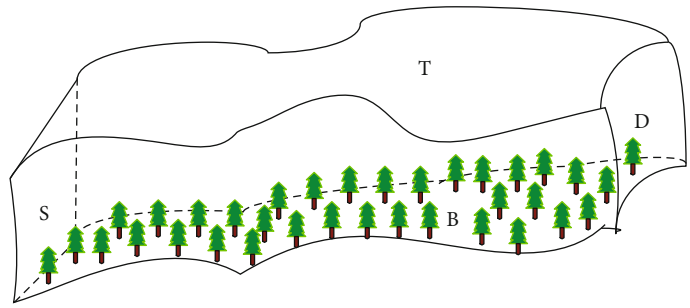


Figure 2: The 3D irregular space model for barrier coverage.

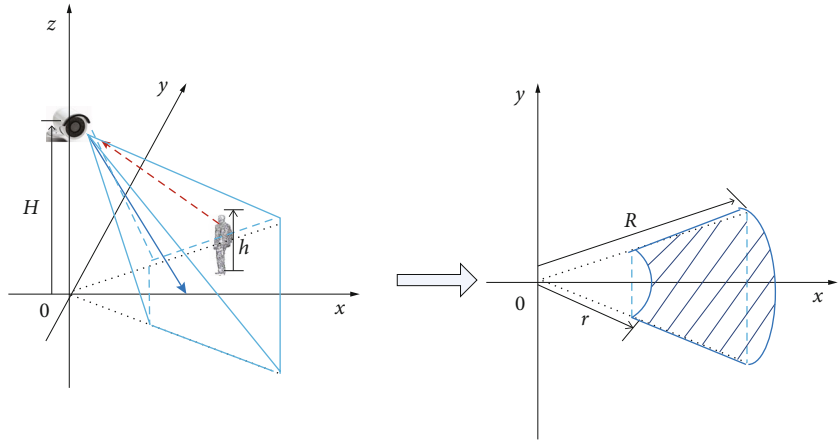


Figure 3: The illustration of camera sensing model.

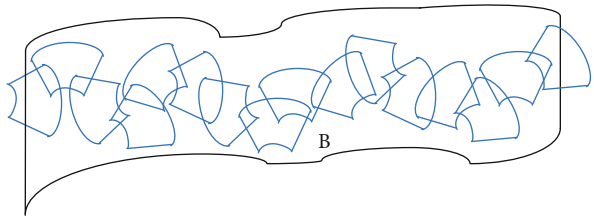


Figure 4: The illustration of network modeling.

sections and $\mathcal{T}$ and $\mathcal{B}$ are the ceiling and the ground planes of $\mathcal{ST}$

(ii) The camera sensor set V deployed in the space $\mathcal{ST}$, $\{v_1, v_2, \cdots, v_N\}$, in which each node v_i has its position (X_i, Y_i, H_i) and maximum working duration l_i

(iii) A potential target or intruder crossing the space $\mathcal{S}$ $\mathcal{T}$ with the predefined height h and face length L

(iv) LifMax-BC Problem is to find a collection of subsets of $VB = \{\text{barrier}_1, \text{barrier}_2, \text{barrier}_3, \cdots\}$ in which each barrier guarantees the strong barrier coverage of the potential target, and schedule these barriers in sleep-and-awake mode with the barrier lifetimes $\{\text{lifetime}_1, \text{lifetime}_2, \text{lifetime}_3, \cdots\}$

(v) The constraint is that each camera sensor cannot be scheduled to exceed its maximum working duration l_i, i.e., $\sum_{\text{barrier}_k \in B} \text{lifetime}_k \cdot x_i^k \le l_i (1 \le i \le N)$, where x_i^k is a binary variable to denote whether v_i is scheduled in barrier_k (if $v_i \in \text{barrier}_k$, $x_i^k = 1$; otherwise $x_i^k = 0$)

(vi) The goal is maximizing the network lifetime $\sum_{\text{barrier}_k \in B} \text{lifetime}_k$

To analyze the hardness of our problem, we review a classical NP-hard problem in graph theory, Minimum Weighted Set Cover (MWSC) Problem, which mathematical formulation is as follows:

Given a set A composed of n elements, a collection C of m subsets of $A (C = \{A_1, A_2, \cdots, A_m\})$ where each $A_j \in C$ $(1 \le j \le m)$ with a weight $w(A_j)$, the problem is to find the minimum weighted subcollection $C_0 \subseteq C$ such that $\bigcup_{A_j \in C_0} A_j = A$ and $\sum_{A_j \in C_0} w(A_j)$ is minimized.

Based on the definition of MWSC Problem, the hardness proof of our problem is given as follows.

Theorem 3. *The LifMax-BC Problem is NP-hard.*

Proof 1. In order to prove the hardness of LifMax-BC Problem, we consider the special case of the problem: each node only contributes to only one barrier, i.e., its working

duration overall contributes to the barrier it belongs to and $\sum_k x_i^k = 1$. Based on the ROI sensing model, we can construct candidate node-disjoint barriers $\{\text{barrier}_1, \text{barrier}_2, \cdots, \text{barrier}_{k'}\}$ on the structural parameters of the space $\mathcal{ST}$ and the known height and face length of the potential target. With the predefined maximum working duration l_i of each sensor, we can calculate the lifetime of each barrier, i.e., $\text{lifetime}_\text{k} = \min_{v_i \in \text{barrier}_k} l_i$. If we assign the inverse of the lifetime as a weight to each barrier, i.e., $\text{weight}(\text{barrier}_k) = 1/\text{lifetime}_k$, we can rewrite the problem in the case with different l_is as follows:

Given a sensor set $V = \{v_1, v_2, \cdots, v_N\}$ and a barrier set $C = \{\text{barrier}_1, \text{barrier}_2, \cdots, \text{barrier}_{K'}\}$ in which each barrier is a subset of V and can guarantee the barrier coverage for the space $\mathcal{ST}$, the problem is to find a subset of C, e.g. $C_0 = \{\text{barrier}_1, \text{barrier}_2, \cdots, \text{barrier}_k\}$, such that $\sum_{k=1}^{K} \text{weight}(\text{barrier}_k)$ is minimized and $\cup_{\text{barrier}_k \in C_0} \text{barrier}_k = V$.

Since the special version of LifMax-BC Problem is equivalent to MWSC Problem which is proven to be NP-hard [20]. Therefore, LifMax-BC Problem is NP-hard in general. □

To solve LifMax-BC Problem, we firstly consider the problem in homogeneous networks (denoted as Homo-LifMax-BC Problem), i.e., the camera sensors have the uniform working duration l_0. We design a barrier coverage scheduling algorithm with disjoint barriers, Robust Barrier Coverage Algorithm. Secondly, we propose the scheduling algorithm with intersecting barriers for the problem in heterogeneous networks (denoted as Hetero-LifMax-BC Problem), i.e., the camera sensors have different maximum working duration l_is, which is called as Enhancing Barrier Coverage Algorithm. The descriptions and analysis of these two algorithms are presented in the next two sections.

4. Robust Barrier Coverage Algorithm for Homo-LifMax-BC Problem

Consider the case of homogeneous camera sensors with the same maximum working duration l_0, the robustness of the network is important. And the pivotal key is avoiding the exhausted situation of some sensor, which will lead to the failure of the barriers that the sensor works for. Thus the scheduling should balance each sensor's function in the coverage barriers. With the goal of maximizing the network lifetime, we adopt the sleep-and-awake mode for scheduling, i.e., there is one barrier working and the other barriers are in sleep mode for each round. The sleep-and-awake mode can transform the original goal into maximizing the number of batches of the constructed coverage barriers. For Homo-LifMax-BC Problem, we design Robust Barrier Coverage Algorithm which is composed of two phases, Auxiliary Graph Transformation and Barrier Coverage Scheduling.

4.1. Auxiliary Graph Transformation in Robust Barrier Coverage. To design a sleep-and-awake scheduling for barrier coverage, the first phase is to give an equivalent transformation for the network model $G = (V, E)$ illustrated in subsection 3.2. The 3D graph has the node set of camera sensors $V = \{v_1, v_2, \cdots, v_N\}$ and is connected by the intersection of the sensing ranges among the sensors, i.e., $\text{SecRing}_i \cap \text{SecRing}_j \neq \varnothing (1 \le i, j \le N)$. For each node in V, we define its neighbor set and degree as $\text{Neighb}(v_i) = \{v_j | v_j \in V \wedge (v_i, v_j) \in E\}$ and $\deg(v_i) = |\text{Neighb}(v_i)|$, respectively. Then, we transform the undirected and unweighted graph G into a directed and edge-weighted graph G^* according to the following steps:

Step 1. Virtual source and destination introducing. To guarantee the strong barrier coverage of any potential target in the space $\mathcal{ST} = (\mathcal{S}, \mathcal{D}, \mathcal{T}, \mathcal{B})$, it is important to construct a consecutive barrier without interval. To the end, we introduce two virtual nodes on the terminal sections $\mathcal{S}$ and $\mathcal{D}$, respectively, i.e., $V \longleftarrow V \cup \{s, t\}$, as shown in Figure 5. For the additional source s on $\mathcal{S}$, we add new edges to connect s and the nodes with the sensing range intersecting with the terminal section $\mathcal{S}$, i.e., $E \longleftarrow E \cup \{(s, v_i) | v_i \in V \wedge \text{SecRing}_i \cap \mathcal{S} \neq \varnothing\}$. For example, v_1's sensing range intersects with $\mathcal{S}$ in G, and then the edge (s, v_1) can be added into E, as shown in Figure 5. In a similar way, for the additional destination d on $\mathcal{D}$, the new edges are added to connect d and the nodes with the sensing range intersecting with the terminal section $\mathcal{D}$, i.e., $E \longleftarrow E \cup \{(v_j, d) | v_j \in V \wedge \mathit{SecRing}_j \cap \mathcal{D} \neq \varnothing\}$. Then, we update the network graph as $G = (V, E)$ by introducing s and t and we construct the auxiliary graph G^* in the next steps.

Step 2. Node-to-directed-edge converting. Based on the updated graph $G = (V, E)$, we give an equivalent transformation for each node v_i in V (with the exception of s and d): v_i is converted into a directed edge $\langle v_i, v'_i \rangle$ with the weight $\text{weight}(\langle v_i, v'_i \rangle) = \deg(v_i)$. For example, v_1 with the degree 3 in the original G corresponds to the directed edge $<v_1, v'_1>$ with $\text{weight}(\langle v_1, v'_1 \rangle) = 3$ in G^* as shown in Figure 5.

Note that there is a clear division of each pair v_i and v'_i on the function of connecting directed edges in Step 3. v_i will be the destination of all the ingoing edges to v_i and v'_i will be the source of all the outgoing edges from v_i in the original G which will give the detailed examples in Step 3. Then for the auxiliary graph G^*, the node set $V^* = V \cup \{v'_i | v_i \in V\}$, the edge set $E^* = \{<v_i, v'_i> | v_i \in V\}$, and the edge-weight set $W^* = \{\mathit{weight}(\langle v_i, v'_i \rangle) | < v_i, v'_i > \in E^*\}$.

Step 3: Undirected-edge-to-directed-edge duplexing. For the auxiliary graph G^*, we will transform the edges in the original G and assign the weights for them, which are considered into the following three cases:

(i) The outgoing edges from s: For each edge (s, v_i) in E, it is converted into a directed edge $\langle s, v_i \rangle$ with the weight $\mathit{weight}(\langle s, v_i \rangle) = 1$. Then $E^* = E^* \cup \{\langle s, v_i \rangle | (s, v_i) \in E\}$ and $W^* = W^* \cup \{\mathit{weight}(\langle s, v_i \rangle) | (s, v_i) \in E\}$. For example, the edge (s, v_1) in G has a corresponding edge $\langle s, v_i \rangle$ with $\mathit{weight}(\langle s, v_i \rangle) = 1$ in G^* as shown in Figure 5.

(ii) The bidirectional edges between (v_i, v_j)s: For each edge (v_i, v_j) in E, it is transformed into two directed

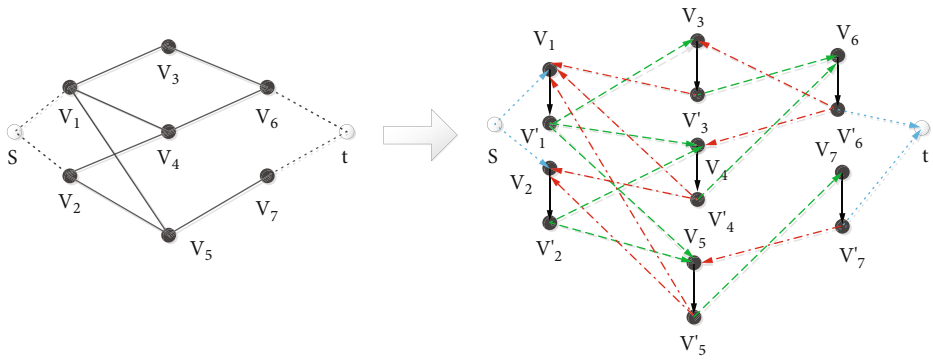


Figure 5: An instance of auxiliary graph transformation.

edges $\langle v'_i, v_j\rangle$ and $\langle v'_j, v_i\rangle$ which both have the weight 1. Then $E^* = E^* \cup \{\langle v'_i, v_j\rangle, \langle v'_j, v_i\rangle | (v_i, v_j) \in E\}$ and $W^* = W^* \cup \{weight(\langle v'_i, v_j\rangle), weight(\langle v'_j, v_i\rangle) | (v_i, v_j) \in E\}$. For example, if there is an edge (v_5, v_7) in G, the directed edges $\langle v'_5, v_7\rangle$ and $\langle v'_7, v_5\rangle$ are connected in G^* as shown in Figure 5.

(iii) The ingoing edges to d: For each edge (v_j, d) in E, it is converted into a directed edge $<v'_j, d>$ with the weight $weight(\langle v'_j, d\rangle) = 1$. Then $E^* = E^* \cup \{\langle v'_j, d\rangle | (v_j, d) \in E\}$ and $W^* = W^* \cup \{weight(\langle v'_j, d\rangle) | (v_j, d) \in E\}$. For example, the edge (v_6, t) in G has a corresponding edge $\langle v'_6, t\rangle$ with $weight(\langle v'_6, t\rangle) = 1$ in G^* as shown in Figure 5.

After the above three steps, we construct an equivalent auxiliary graph $G^* = (V^*, E^*, W^*)$ for the next barrier constructing in subsection 4.2.

4.2. Barrier Coverage Scheduling in Robust Barrier Coverage. Based on $G^* = (V^*, E^*, W^*)$, we construct the feasible barriers with the maximum number based on the maximum flow algorithm as follows.

Firstly, by taking G^*, s and t as the source and destination as the input of the Stint Algorithm in [21], K node-disjoint flows can be obtained which can be restored into a path set $\mathscr{P} = \{\text{path}_1, \text{path}_2, \cdots, \text{path}_K\}$. The paths in $\mathscr{P}$ are node-disjoint which can satisfy the requirement of LifMax-BC Problem in homogeneous network, i.e., it can avoid the situation that some sensor's expiration causes the failure of the barriers which this sensor works for.

Secondly, since the path set is constructed in the auxiliary graph G^*, we need to reduce these paths in G^* back into the barriers in G. Here we take one path as an example to explain the reduction process and the reduction of other paths are in the same way. For each $path_k = \{s, v_1, v'_1, v_2, v_2', \cdots, v_p, v'_p, d\}$ in $\mathscr{P}(1 \le k \le K)$, it can be reduced back to a coverage barrier $barrier_k$ according to the following process: for the edge $\langle s, v_i\rangle$, it is restored into the source v_i of $barrier_k$ in G; for the edge $\langle v'_i, v_j\rangle$, it is reduced into the undirected edge (v_i, v_j) of $barrier_k$ in G; for the edge $\langle v'_j, d\rangle$, it is restored into the destination v_j of $barrier_k$ in G. As shown in Figure 6, two node-disjoint paths in the auxiliary graph are reduced into two barriers $\{v_1, v_3, v_6\}$ and $\{v_2, v_5, v_7\}$. And the lifetime of each barrier is l_0 in the homogeneous network, thus the network lifetime is $K \cdot l_0$.

The detailed description of Robust Barrier Coverage Algorithm for Homo-LifMax-BC Problem is given in Algorithm 1.

To analyze the time complexity of Algorithm 1, we review the two phases of the algorithm: for auxiliary graph transformation, the virtual nodes introducing, node converting, and edge duplexing take the time of $O(1)$, $O(|V|)$ and $O(|E|)$, respectively. Thus the time complexity of Phase 1 is $O(|V|)$ under the assumption that $|V| > |E|$ in barrier coverge; for barrier coverage scheduling, it adopts the Stint Algorithm for constructing the node-disjoint flows with the maximum number, which has the time complexity of $O(|V| \cdot |E|^2)$. And the reduction process takes the time of $O(|V| \cdot |E|)$. To sum up, the time complexity of Algorithm 1 is $O(|V| \cdot |E|^2)$.

5. Enhancing Barrier Coverage Algorithm for Hetero-LifMax-BC Problem

Consider the heterogeneous camera sensors with the different maximum working duration l_is, increasing the coverage efficiency of the sensors with high l_is is beneficial to maximize the network lifetime. If applying Algorithm 1 to solve Hetero-LifMax-BC Problem, disjoint barriers will lead to the situation that there is plenty of extended sleep time for the sensors with high l_is, which goes against the problem goal of maximizing network lifetime. Thus it is necessary to design another algorithm to efficiently schedule the sensors with high durations from the perspective of making contribution to as many barriers as possible. Then, we design Enhancing Barrier Coverage Algorithm for Hetero-LifMax-BC Problem, whose main idea is to schedule the sensors with longer working durations and higher degrees in more barriers to maximize the network lifetime. There are also two phases in this algorithm, Auxiliary Graph Transformation and Barrier Coverage Scheduling.

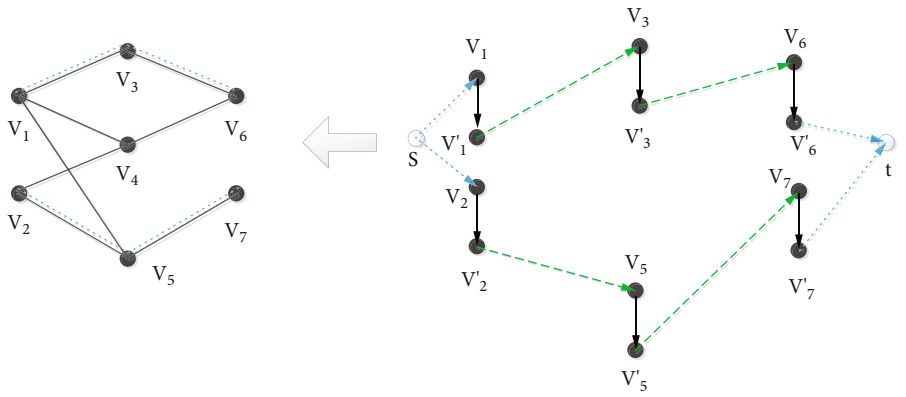


Figure 6: An instance of barrier reduction.

1: Set $V^*, E^*, W^* \longleftarrow \varnothing$, $Lifetime = 0$
2: **for** each sensor v_i in V **do**
3: $Neighb(v_i) = \{v_j | v_j \in V \wedge (v_i, v_j) \in E\}$, $\deg(v_i) = |Neighb(v_i)|$
4: **//Phase 1: Auxiliary Graph Transformation**
5: **//Step 1: Virtual source and destination introducing**
6: $V \longleftarrow V \cup \{s, t\}$
7: $E \longleftarrow E \cup \{(s, v_i), (v_j, d) | v_i, v_j \in V \wedge SecRing_i \cap \mathcal{S} \neq \varnothing \wedge SecRing_j \cap \mathcal{D} \neq \varnothing\}$
8: **//Step 2: Node-to-directed-edge converting**
9: $V^* = V \cup \{v_i' | v_i \in V\}$
10: $E^* = \{\langle v_i, v_i' \rangle | v_i \in V\}$
11: $W^* = \{weight(\langle v_i, v_i' \rangle) = \deg(v_i) | \langle v_i, v_i' \rangle \in E^*\}$
12: **//Step 3: Undirected-edge-to-directed-edge duplexing**
13: **for** each edge (s, v_i) in E **do**
14: $E^* = E^* \cup \{\langle s, v_i \rangle\}$, $W^* = W^* \cup \{weight(\langle s, v_i \rangle) = 1\}$
15: **for** each edge (v_i, v_j) in E **do**
16: $E^* = E^* \cup \{\langle v_i', v_j \rangle, \langle v_j', v_i \rangle\}$, $W^* = W^* \cup \{weight(\langle v_i', v_j \rangle) = 1, weight(\langle v_j', v_i \rangle) = 1\}$
17: **for** each edge (v_j, d) in E **do**
18: $E^* = E^* \cup \{\langle v_j', d \rangle\}$, $W^* = W^* \cup \{weight(\langle v_j', d \rangle) = 1\}$
19: $G^* = (V^*, E^*, W^*)$
20: **//Phase 2: Barrier Coverage Scheduling**
21: Apply Stint Algorithm in [21] to (G^*, s, t) and obtain K node-disjoint paths collected in $\mathcal{P} = \{path_1, path_2, \cdots, path_K\}$
21: **for** each path $path_k$ in $\mathcal{P}$ **do**
23: **for** each directed edge on $path_k$ **do**
24: Case 1: for $\langle s, v_i \rangle$, it is restored into the source v_i of $barrier_k$.
25: Case 2: for each $\langle v_i', v_j \rangle$, it is reduced into the undirected edge (v_i, v_j) of $barrier_k$.
26: Case 3: for $\langle v_i', d \rangle$, it is restored into the destination v_j of $barrier_k$.
27: $Lifetime = K \cdot l_0$
28: $\{barrier_1, barrier_2, \cdots, barrier_K\}$, $Lifetime$.

Algorithm 1: Robust Barrier Coverage Algorithm for Homo-LifMax-BC Problem $(\mathcal{SFT} = (\mathcal{S}, \mathcal{D}, \mathcal{T}, \mathcal{B}), \boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}))$

5.1. Auxiliary Graph Transformation in Enhancing Barrier Coverage. This phase is composed of three steps, and we adopt the same Step 1. Virtual source and destination introducing as that in Section 4 and obtain the updated graph $G = (V, E)$.

To construct an auxiliary graph for barrier scheduling in the heterogeneous network, it is necessary to balance two important parameters for each node: the maximum working duration l_i and the neighborhood scale $\deg(v_i)$. To this end, we introduce a new measure for each node

as lifdeg$(v_i) = l_i/\deg(v_i)$, which represents the node's possible average working duration for each neighbor. Furthermore, to analyze each node's contribution for barrier coverage, we need to give the criterion for parameters l_i and $\deg(v_i)$. (1) For the node degree, v_i's degree is identified as **high-degree** if $\deg(v_i) > 1$; otherwise, it is regarded as low-degree. (2) For the maximum working duration, l_i is identified as high-lifetime if $l_i \geq AvgL$, where $AvgL = \sum_{1 \leq i \leq N} l_i/N$; otherwise, it is regarded as low-lifetime. Based on the above preliminaries, we explain the process for Step 2 as follows.

Step 2. Node-to-directed-edge converting. Based on the graph $G = (V, E)$ added with s and t, the transformation for the nodes in V (with the exception of s and d) is divided into the following three cases:

(i) $V_1 = \{v_i | v_i \text{ is low} - \text{degree}\}$. Since $\deg(v_i) = 1$ which stands for that v_i has only one neighbor, v_i will contribute on only one barrier if v_i is scheduled in Phase 2, which is regardless of whether v_i is high-lifetime or low-lifetime. In this case, $\boldsymbol{v_i}$ is converted into 1 directed edge $<v_i, v'_i>$ with the weight $weight(<v_i, v'_i>) = l_i$

(ii) $V_2 = \{v_i | v_i \text{ is high} - \text{degree and low} - \text{lifetime}\}$. In this case, v_i will give high contribution for several barriers and we divide its working duration equally to each connectivity relationship. In details, v_i is converted into $\deg(v_i)$ directed edges $\langle v_i^d, v'^d_i \rangle$s $(1 \leq d \leq \deg(v_i))$ with the uniform weight weight$(\langle v_i^d, v'^d_i \rangle) = \text{lifdeg}(v_i)$

(iii) $V_3 = \{v_i | v_i \text{ is high} - \text{degree and low} - \text{lifetime}\}$. This case is the most complicated because of its possible unbalanced contributions to several barriers in scheduling. To avoid the unbalance, we propose a trade-off approach to guarantee the reasonably efficient scheduling for such nodes.

Firstly, we sort v_i's neighbors in nonincreasing order on the values of lifdeg(v_j), where $v_j \in Neighb(v_i)$. Secondly, we calculate the maximum value of the sum of the first $\text{sum}(v_i)$ neighbors' lifdeg(v_j)s, which is no more than l_i. In other words, we maximize the contribution of v_i on a part of neighbors (the first $\text{sum}(v_i)$ neighbors) rather than all the neighbors (Neighb(v_i)). Then we update Neighb(v_i) as $\{\text{neighb}_i^1, \text{neighb}_i^2, \cdots, \text{neighb}_i^{\text{sum}(v_i)}\}$ by only retaining the first $\text{sum}(v_i)$ neighbors and eliminating other neighbors. Then, $\deg(v_i) = \text{sum}(v_i)$. Thirdly, we perform the transformation of v_i: v_i is converted into $\text{sum}(v_i)$ directed edges $\langle v_i^u, v'^u_i \rangle$ with the different weights weight$(\langle v_i^u, v'^u_i \rangle) = \text{lifdeg}(\text{neighb}_i^u)(1 \leq u \leq \text{sum}(v_i))$.

To conclude the above three cases, there is also a clear division of each pair v_i and v'_i on the function of connecting directed edges in Step 3. v_i is in charge of all the ingoing edges to v_i and v'_i is responsible for all the outgoing edges from v_i in the original G. And the construction of the auxiliary graph G^* in Step 2 is as follows:

(a) The node set $V^* = \bigcup_{v_i \in V_1} \{v_i, v'_i\} \cup \bigcup_{v_i \in V_2} \{v_i^d, v'^d_i | 1 \leq d \leq \deg(v_i)\} \cup \bigcup_{v_i \in V_3} \{v_i^u, v'^u_i | 1 \leq u \leq \text{sum}(v_i)\}$

(b) The edge set $E^* = \bigcup_{v_i \in V_1} \{<v_i, v'_i>\} \cup \bigcup_{v_i \in V_2} \{<v_i^d, v'^d_i> | 1 \leq d \leq \deg(v_i)\} \cup \bigcup_{v_i \in V_3} \{<v_i^u, v'^u_i> | 1 \leq u \leq \text{sum}(v_i)\}$

(c) The edge-weight set $W^* = \{\text{weight}(\langle v_i, v'_i \rangle) | <v_i, v'_i> \in E^*\}$

Step 3. Undirected-edge-to-directed-edge duplexing. Based on the partial of G^* constructed in Step 2, we will add new directed edges by transforming the undirected edges in the original G as follows:

(i) The outgoing edges from s are as follows: for each edge (s, v_i) in E, it is converted into a directed edge $\langle s, v_i \rangle$ with weight$(\langle s, v_i \rangle) = 0$, if $v_i \in V_1$; it is converted into $\deg(v_i)$ directed edges $\langle s, v_i^d \rangle$ with weight$(\langle s, v_i^d \rangle) = 0$, if $v_i \in V_2$; it is converted into $\text{sum}(v_i)$ directed edges $\langle s, v_i^u \rangle$ with weight$(\langle s, v_i^u \rangle) = 0$, if $v_i \in V_3$. Then $E^* = E^* \cup \bigcup_{v_i \in V_1 \wedge (s,v_i) \in E} \{<s, v_i>\} \cup \bigcup_{v_i \in V_2 \wedge (s,v_i) \in E} \{<s, v_i^d> | 1 \leq d \leq \deg(v_i)\} \cup \bigcup_{v_i \in V_3 \wedge (s,v_i) \in E} \{<s, v_i^u> | 1 \leq u \leq \text{sum}(v_i)\}$ and $W^* = W^* \cup \{weight(\langle s, v_i \rangle) | <s, v_i> \in E^*\}$

(ii) The bidirectional edges between (v_i, v_j)s: For each edge (v_i, v_j) in E, it is transformed into $\deg(v_i) \cdot \deg(v_j)$ pairs of directed edges $\langle v'^{d_i}_i, v_j^{d_j} \rangle$ and $\langle v'^{d_j}_j, v_i^{d_i} \rangle (1 \leq d_i \leq \deg(v_i)$ and $1 \leq d_j \leq \deg(v_j))$ which all have the weight 0. Then $E^* = E^* \cup \bigcup_{(v_i,v_j) \in E} \{\langle v'^{d_i}_i, v_j^{d_j} \rangle, \langle v'^{d_j}_j, v_i^{d_i} \rangle | 1 \leq d_i \leq \deg(v_i)$ and $1 \leq d_j \leq \deg(v_j)\}$ and $W^* = W^* \cup \{\text{weight}(\langle v'_i, v_j \rangle), \text{weight}(\langle v'_j, v_i \rangle) | \langle v'_i, v_j \rangle, \langle v'_j, v_i \rangle \in E^*\}$.

(iii) The ingoing edges to d are as follows: For each edge (v_j, d) in E, the transformation is similar with that of (i), i.e., $E^* = E^* \cup \bigcup_{v_j \in V_1 \wedge (v_j,d) \in E} \{\langle v'_j, d \rangle\} \cup \bigcup_{v_j \in V_2 \wedge (v_j,d) \in E} \{\langle v'^d_j, d \rangle | 1 \leq d \leq \deg(v_j)\} \cup \bigcup_{v_j \in V_3 \wedge (v_j,d) \in E} \{\langle v'^u_j, d \rangle | 1 \leq u \leq \text{sum}(v_j)\}$ and $W^* = W^* \cup \{\text{weight}(\langle v'_j, d \rangle) = 0 | <v'_j, d> \in E^*\}$

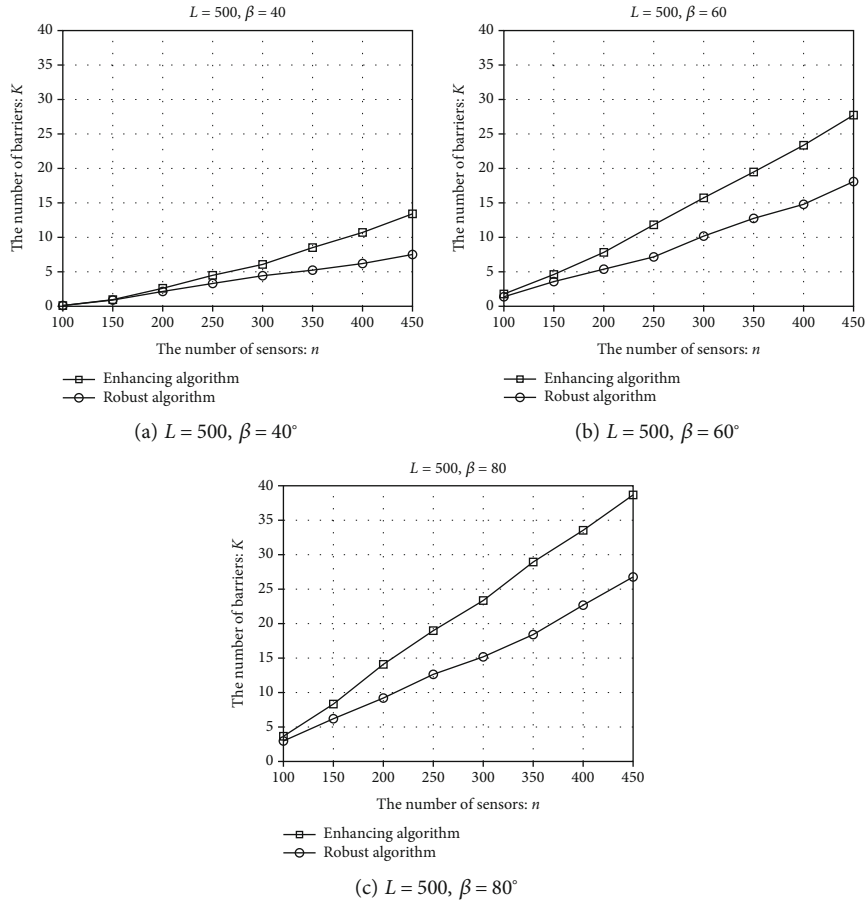
5.2. Barrier Coverage Scheduling in Enhancing Barrier Coverage. For this phase, we input the constructed auxiliary graph (G^*, s, t) to Stint Algorithm [21] and generate K node-disjoint flows which are collected in $\mathcal{P} = \{path_1, path_2, \cdots, path_K\}$. Note that since we divide the nodes with high possible contributions into several independent directed edges in Step 2 of Phase 2, we can also apply the node-

Set $V^*, E^*, W^* \longleftarrow \varnothing$, $Lifetime = 0$, $AvgL = \sum_{1 \le i \le N} l_i / N$
for each sensor v_i in V **do**
  $Neighb(v_i) = \{v_j | v_j \in V \wedge (v_i, v_j) \in E\}$, $\deg(v_i) = |Neighb(v_i)|$
  $lifdeg(v_i) = l_i / \deg(v_i)$
//**Phase 1: Auxiliary Graph Transformation**
Set $V_1, V_2, V_3 \longleftarrow \varnothing$
for each sensor v_i in V **do**
  Case 1: if $\deg(v_i) = 1$, $V_1 \longleftarrow V_1 \cup \{v_i\}$
  Case 2: if $\deg(v_i) > 1$ and $l_i \ge AvgL$, $V_2 \longleftarrow V_2 \cup \{v_i\}$
  Case 3: if $\deg(v_i) > 1$ and $l_i < AvgL$, $V_3 \longleftarrow V_3 \cup \{v_i\}$
//**Step 1: Virtual source and destination introducing**
$V \longleftarrow V \cup \{s, t\}$
$E \longleftarrow E \cup \{(s, v_i), (v_j, d) | v_i, v_j \in V \wedge SecRing_i \cap \mathcal{S} \ne \varnothing \wedge SecRing_j \cap \mathcal{D} \ne \varnothing\}$
//**Step 2: Node-to-directed-edge converting**
for each sensor v_i in V_1 **do**
  $V^* = V^* \cup \{v_i, v_i'\}$, $E^* = E^* \cup \{\langle v_i, v_i' \rangle\}$, $W^* = W^* \cup \{weight(\langle v_i, v_i' \rangle) = l_i\}$
for each sensor v_i in V_2 **do**
  $V^* = V^* \cup \{v_i^d, v'^d_i | 1 \le d \le \deg(v_i)\}$, $E^* = E^* \cup \{<v_i^d, v'^d_i> | 1 \le d \le \deg(v_i)\}$, $W^* = W^* \cup \{weight(<v_i^d, v'^d_i>) = lifdeg(v_i) | 1 \le d \le \deg(v_i)\}$
for each sensor v_i in V_3 **do**
  Calculate the maximum value of the sum of the first $sum(v_i)$ neigh-bors' $lifdeg(v_j)$s, which is no more than l_i.
  $Neighb(v_i) = \{neighb_i^1, neighb_i^2, \cdots, neighb_i^{sum(v_i)}\}$, $\deg(v_i) = |Neighb(v_i)|$
  $V^* = V^* \cup \{v_i^u, v'^u_i | 1 \le u \le sum(v_i)\}$, $E^* = E^* \cup \{<v_i^u, v'^u_i> | 1 \le u \le sum(v_i)\}$, $W^* = W^* \cup \{weight(<v_i^u, v'^u_i>) = lifdeg(neighb_i^u) | 1 \le u \le sum(v_i)\}$
//**Step 3: Undirected-edge-to-directed-edge duplexing**
for each edge (s, v_i) in E **do**
$E^* = E^* \cup \bigcup_{v_i \in V_1 \wedge (s, v_i) \in E} \{<s, v_i>\} \cup \bigcup_{v_i \in V_2 \cup V_3 \wedge (s, v_i) \in E} \{\langle s, v_i^d \rangle | 1 \le d \le \deg(v_i)\}$, $W^* = W^* \cup \{weight(\langle s, v_i \rangle) = 0 | < s, v_i > \in E^*\}$
for each edge (v_i, v_j) in E **do**
$E^* = E^* \cup \bigcup_{(v_i, v_j) \in E} \{\langle v'^{d_i}_i, v_j^{d_j} \rangle, \langle v'^{d_j}_j, v_i^{d_i} \rangle | 1 \le d_i \le \deg(v_i) \text{ and } 1 \le d_j \le \deg(v_j)\}$, $W^* = W^* \cup \{weight(\langle v_i', v_j \rangle) = 0, weight(\langle v_j', v_i \rangle) = 0 | < v_i', v_j >, < v_j', v_i > \in E^*\}$
for each edge (v_j, d) in E **do**
$E^* = E^* \cup \bigcup_{v_j \in V_1 \wedge (v_j, d) \in E} \{\langle v_j', d \rangle\} \cup \bigcup_{v_j \in V_2 \cup V_3 \wedge (v_j, d) \in E} \{v'^d_j, d | 1 \le d \le \deg(v_j)\}$, $W^* = W^* \cup \{weight(\langle v_j', d \rangle) = 0 | < v_j', d > \in E^*\}$
$G^* = (V^*, E^*, W^*)$
//**Phase 2: Barrier Coverage Scheduling**
Apply the maximum flow algorithm in [21] to (G^*, s, t) and obtain K node-disjoint paths collected in $\mathcal{P} = \{path_1, path_2, \cdots, path_K\}$
for each path $path_k$ in $\mathcal{P}$ **do**
  for each directed edge on $path_k$ **do**
    Case 1: for $\langle s, v_i \rangle$ or $\langle s, v_i^d \rangle$, it is restored into the source v_i of $barrier_k$.
    Case 2: For each $\langle v'_i, v_j \rangle$ or $\langle v'^{d_i}_i, v_j^{d_j} \rangle$, It Is reduced into the undirected edge (v_i, v_j) of $barrier_k$.
    Case 3: For $\langle v'_j, d \rangle$ or $\langle v'^d_j, d \rangle$, It Is Restored into the Destination v_j of $barrier_k$.
  $lifetime_k = \min_{<v_i, v_i'> \in path_k} weight(\langle v_i, v_i' \rangle)$
$Lifetime = \sum_{1 \le k \le K} lifetime_k$
Return $\{barrier_1, barrier_2, \cdots, barrier_K\}$, $Lifetime$.

Algorithm 2: Enhancing Barrier Coverage Algorithm for Hetero-LifMax-BC Problem $(\mathcal{SDTB}) = (\mathcal{S}, \mathcal{D}, \mathcal{T}, \mathcal{B}), \boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$

disjoint flows algorithm, which can realize the scheduling of such nodes in different barriers.

These node-disjoint paths in $\mathcal{P}$ are constructed in the auxiliary graph G^* which are needed to be reduced back into the barriers in G. Since the nodes in V_2 or V_3 may be scheduled in multiply flows in $\mathcal{P}$, the reduction process is different from that of Algorithm 1. For each path $path_k$ in $\mathcal{P}(1 \le k \le K)$, it can be reduced back to a coverage barrier $barrier_k$ based on three kinds of edges: for the edge $\langle s, v_i \rangle$ or $\langle s, v_i^d \rangle$, it is restored into the source v_i of $barrier_k$ in G; for the edge $\langle v'_i, v_j \rangle$ or $\langle v'^{d_i}_i, v_j^{d_j} \rangle$, it is reduced into the undirected edge (v_i, v_j) of $barrier_k$ in G; and for the edge $\langle v'_j, d \rangle$ or $\langle v'^d_j, d \rangle$, it is restored into the destination v_j of $barrier_k$ in G. Finally, the minimum nonzero weight on the corresponding path is the lifetime of each



(a) $L = 500$, $\beta = 40^\circ$

(b) $L = 500$, $\beta = 60^\circ$

(c) $L = 500$, $\beta = 80^\circ$

FIGURE 7: The number of Barriers K vs. number of nodes n.

barrier in the heterogeneous network, i.e., $\text{lifetime}_k = \min_{<v_i, v_i'> \in \text{path}_k} \text{weight}(\langle v_i, v_i' \rangle)$.

The whole description of Enhancing Barrier Coverage Algorithm for LifMax-BC Problem is given in Algorithm 2.

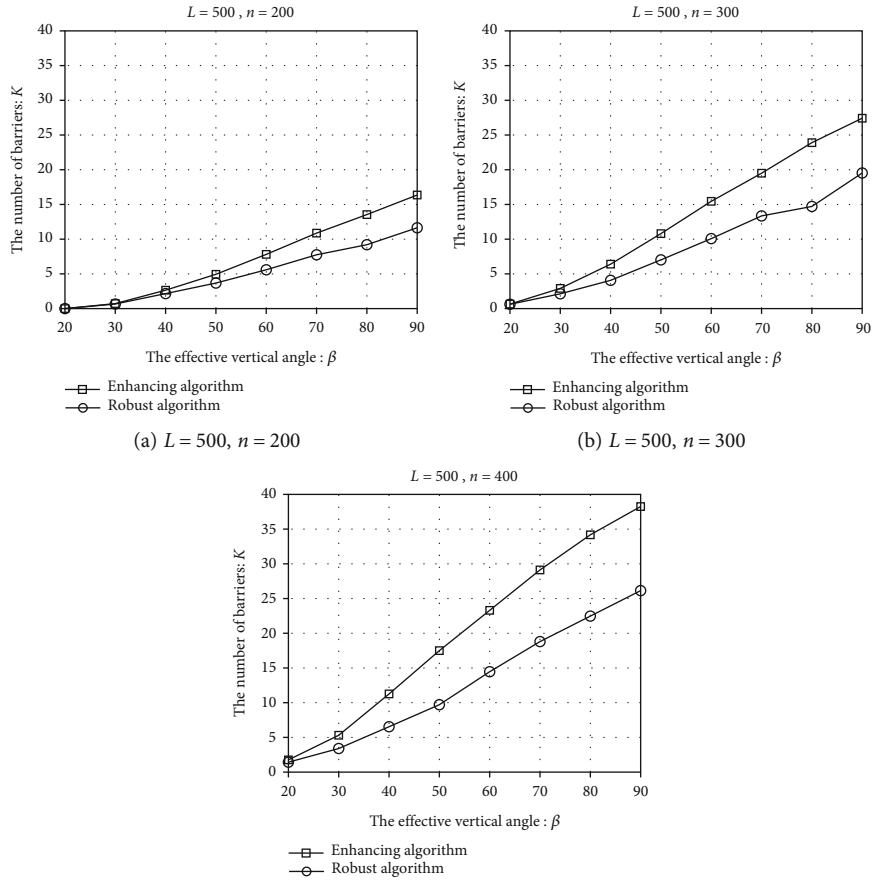
Similarly with the analysis of Algorithm 1, Algorithm 2 also has the time complexity of $O(|V| \cdot |E|^2)$. Thus the running times of our strategies are both polynomial. They are the feasible solutions of LifMax-BC Problem for homogeneous networks and heterogeneous networks.

6. Performance Evaluation

6.1. Experiment Plan. To evaluate the performance of the proposed algorithms for LifMax-BC Problem, we perform a series of experiments to compare their performance by JAVA. The optimization goal of LifMax-BC Problem is maximizing the network lifetime and we solve it for two cases (the same working duration and the different ones). Instead of the network lifetime, we choose the number of barriers K as the evaluation criterion. It is because that the number of constructed barriers stands for the number of scheduling rounds, which is more objective and fairer than the length of network lifetime, especially in the case that there is a big difference in the sensors' working duration. Here, we denote the two algorithms as Robust Algorithm and Enhancing Algorithm for short.

The experiments are performed in an irregular 3D space which is a 3D curve strip space with the length of 500 units, the width of 300 units, and the height in the range of $[50, 80]$ units, i.e., the ceiling plane of the space is an irregular curved surface. And the boundary located in the space has the length of L. For the camera sensor network, n camera sensors are randomly deployed on the ceiling plane of the space, i.e., their positions (X_i, Y_i, H_i) are randomly valued in the scope of the space. And the process of deployment is successfully finished when the network graph is connected. For each camera sensor, it has the sensing radius of 100 units, the Field-of-Vision 60° and the effective vertical angle β. And the maximum working duration l_i of each sensor is uniformed as 10 for Robust Algorithm and valued in the range of $[5, 30]$ for Enhancing Algorithm. Based on the ROI sensing model, each sensor has the minimum effective vertical angle $\beta_{\min} = 0^\circ$ and the maximum effective vertical angle $\beta_{\max} = \beta$; for the potential target/intruder, we set the height as 17 units and the face length as 2 units for general situations.

In the experiments, we will investigate the performance of the scheduling algorithms from two important parameters: the number of camera sensors n and the effective vertical angle β, which are corresponding to two groups of



(a) $L = 500$, $n = 200$

(b) $L = 500$, $n = 300$

(c) $L = 500$, $n = 400$

FIGURE 8: The number of barriers K vs. effective vertical angle β.

settings: Group 1—n varies from 100 to 450 by the step of 50 (a) $L = 500$, $\beta = 40°$; (b) $L = 500$, $\beta = 60°$; and (c) $L = 500$, $\theta = 80°$. Group 2—β varies from 20° to 90° by the step of 1 0° (a) $L = 500$, $n = 200$; (b) $L = 500$, $n = 300$; and (c) $L = 500$ $n = 400$. For each parameter setting, we run 100 instances and compute their average for evaluation.

6.2. Experiment Result Analysis. As the results in Figure 7 shown, it can be observed that the number of constructed barriers from the proposed algorithms present rising trend with the enlargement of the networks n. Between the two algorithms, Enhancing Algorithm is more influenced by n, i.e., K obtained by the algorithm grows faster with the increasing of n; Robust Algorithm is less effected by n, and the gap between the results from the two algorithms becomes larger with the growth of n. The increasing of the network scale can satisfy more requirements of strong barrier coverage and Enhancing Algorithm utilizes some sensor for multiple barriers, which increases the number of barriers. Furthermore, from Figures 7(a) and 7(c), the better performance of Enhancing Algorithm becomes more significant when the effective vertical angle $\beta = 80°$. It can be explained that the enlargement of the effective vertical angle improves sensors' coverage range which increases the probability of barrier coverage.

Investigating the effect of the effective vertical angles on scheduling algorithms in Figure 8, we can find that β's change has less significant influence when network scale is relatively small and the number of barriers increases slowly in Figures 8(a) and 8(b), while β has more influence on the algorithms' performance from Figure 8(c). Furthermore, the gap between the results from the two algorithms presents smaller than that obtained by varying the number of sensors. Seen from these three subfigures, Robust Algorithm's results presents the rising trend with the increasing of β, which is less than that presented by Enhancing Algorithm. It can be concluded that the sensing conditions have less influence on the coverage quality when the network is homogeneous, and the difference on the coverage range is more beneficial for enhancing the coverage efficiency in the heterogeneous network.

From the above two groups of experiment results, we can conclude that the proposed algorithms are both efficient on maximizing the network lifetime and can be utilized into the solution for LifMax-BC Problem in the homogeneous and heterogeneous networks.

7. Conclusions

In this paper, we investigated the camera sensor scheduling problem for strong barrier coverage in 3D CSNs with the

goal of maximizing the network lifetime, LifMax-BC Problem. The problem has been considered and analyzed for homogeneous networks (all the sensors have the uniform working duration) and heterogeneous networks (the sensors have different working durations). Based on the ROI sensing model, we, respectively, proposed two heuristic algorithms via the auxiliary graph construction and the maximum flow algorithm. The algorithm for homogeneous networks aims to increase the network robustness by constructing the disjoint barriers; and the algorithm for heterogeneous networks realizes the lifetime maximization via enhancing the utilization of the sensors with high working duration. By evaluating the performance of the proposed algorithms, the simulation results were analyzed in terms of the number of the scheduled barriers, which show that the algorithms have high efficiency on maximizing the network lifetime and can adapt to different network types. We will design the distributed strategies for the related optimization problems in the future.

Data Availability

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

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China under Grant (62002022 and 62202054) and the Fundamental Research Funds for the Central Universities (No. BLX201921, No. 2021ZY88).

References

[1] S. Cheng, Z. Cai, J. Li, and H. Gao, "Extracting kernel dataset from big sensory data in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 4, pp. 813–827, 2017.

[2] S. Cheng, Z. Cai, and J. Li, "Curve query processing in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5198–5209, 2015.

[3] Y. Hong, Y. Wang, Y. Zhu, D. Li, Z. Chen, and J. Li, "3D camera sensor scheduling algorithms for indoor multi-objective tracking," *Journal of Combinatorial Optimization*, vol. 39, no. 3, pp. 899–914, 2020.

[4] Y. Wang and G. Cao, "On full-view coverage in camera sensor networks," in *Proceedings of IEEE INFOCOM*, pp. 1781–1789, Shanghai, China, 2011.

[5] P. Si, W. Chengdong, Y. Zhang, Z. Jia, P. Ji, and H. Chu, "Barrier coverage for 3D camera sensor networks," *Sensors*, vol. 17, no. 8, p. 1771, 2017.

[6] Z. He, Z. Cai, S. Cheng, and X. Wang, "Approximate aggregation for tracking quantiles and range countings in wireless sensor networks," *Theoretical Computer Science*, vol. 607, no. 3, pp. 381–390, 2015.

[7] J. Li, S. Cheng, Z. Cai, Y. Jiguo, C. Wang, and Y. Li, "Approximate holistic aggregation in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 13, no. 2, pp. 1–24, 2017.

[8] Y. Wang and G. Cao, "Barrier coverage in camera sensor networks," in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 1–10, Paris, France, 2011.

[9] H. Ma, M. Yang, D. Li, Y. Hong, and W. Chen, "Minimum camera barrier coverage in wireless camera sensor networks," in *2012 Proceedings IEEE INFOCOM*, p. 217, Orlando, FL, 2012.

[10] M. Khanjary, M. Sabaei, and M. R. Meybodi, "Barrier coverage in adjustable-orientation directional sensor networks: a learning automata approach," *Computers and Electrical Engineering*, vol. 72, pp. 859–876, 2018.

[11] Q. Zhang, S. He, and J. Chen, "Toward optimal orientation scheduling for full-view coverage in camera sensor networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Washington, DC, USA, 2016.

[12] L. Li, B. Zhang, and J. Zheng, "A study on one-dimensionalk-coverage problem in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 13, no. 1, 11 pages, 2013.

[13] J. Tian, W. Zhang, G. Wang, and X. Gao, "2D k-barrier duty-cycle scheduling for intruder detection in wireless sensor networks," *Computer Communications*, vol. 43, no. 5, pp. 31–42, 2014.

[14] H. Ma, D. Li, W. Chen, Q. Zhu, and H. Yang, "Energy efficient k-barrier coverage in limited mobile wireless sensor networks," *Computer Communications*, vol. 35, no. 14, pp. 1749–1758, 2012.

[15] D.-S. B. J. Wen, J. Jiang, and W.-H. Dou, "Constructing *k*-barrier coverage in mobile wireless sensor networks," *Journal of Software*, vol. 22, no. 9, pp. 2089–2103, 2011.

[16] X. Liu, B. Yang, and G. Chen, "Full-view barrier coverage in mobile camera sensor networks," *Wireless Networks*, vol. 25, no. 8, pp. 4773–4784, 2019.

[17] Y. Zhu, M. Mei, and Z. Zheng, "Scheduling algorithms for k-barrier coverage to improve transmission efficiency in WSNs," *Multimedia Tools and Applications*, vol. 79, no. 15-16, pp. 10505–10518, 2020.

[18] J. Luo and S. Zou, "Strong k-barrier coverage for one-way intruders detection in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 12, no. 6, Article ID 3807824, 2016.

[19] X. Fan, S. Wang, Y. Wang, X. Jinshan, and K. Chi, "Energy-efficient barrier lifetime prolonging scheme based on repairing in directional sensor networks," *IEEE Systems Journal*, vol. 14, no. 4, pp. 4943–4954, 2020.

[20] M. R. Garey and D. S. Johnson, ""Strong" NP-completeness results," *Journal of ACM*, vol. 25, no. 3, pp. 499–508, 1978.

[21] S. Kumar, T. H. Lai, M. E. Posner, and P. Sinha, "Maximizing the lifetime of a barrier of wireless sensors," *IEEE Transactions on Mobile Computing*, vol. 9, no. 8, pp. 1161–1172, 2010.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1384405, 17 pages
https://doi.org/10.1155/2022/1384405

Research Article

An SKP-ABE Scheme for Secure and Efficient Data Sharing in Cloud Environments

Yong-Woon Hwang,[1] **Su-Hyun Kim**,[2] **Daehee Seo**,[3] **and Im-Yeong Lee**[1]

[1]*Department of Software Convergence, Soonchunhyang University, Asan 31538, Republic of Korea*
[2]*ICT Industry Strategy Team, National IT Industry Promotion Agency, Jincheon-Gun 27872, Republic of Korea*
[3]*Faculty of Artificial Intelligence and Data Engineering, Sangmyung University, Seoul 03016, Republic of Korea*

Correspondence should be addressed to Im-Yeong Lee; imylee@sch.ac.kr

Received 8 April 2022; Revised 15 May 2022; Accepted 20 May 2022; Published 17 June 2022

Academic Editor: Yan Huang

Security threats such as data forgery and leakage may occur when sharing data in cloud environments. Therefore, it is important to encrypt your data and securely access it when sharing it with other users via a cloud server. Of the various security technologies, research on secure data sharing commonly employs Key Policy Attribute-Based Encryption (KP-ABE). However, existing KP-ABE schemes generally lack ciphertext search features. Furthermore, even if a KP-ABE scheme incorporates it, the number of searches required increases markedly by the number of attributes used in the search. It in turn proportionally increases the ciphertext size. In addition, the attribute authority (AA) could be attacked, which can result in the leakage of users' decryption keys. AA is a server that manages user attributes and decryption keys when using attribute-based encryption in a cloud environment. If the AA is curious, it can cause problems with the key escrow with the attributes and decryption (secret) key information of the users it knows. In this paper, to solve all these problems, we present a new scheme called Searchable Key-Policy Attribute-Based Encryption (SKP-ABE) for secure and efficient data sharing in the cloud. This proposed SKP-ABE scheme allows fast ciphertext search and keeps the ciphertext of constant size. The key escrow problem is solved via user key generation.

1. Introduction

Developments in cloud computing technology have made it possible to collect, manage, and share big data from the Internet of Things (IoT)-Cloud environments such as Unmanned Traffic Management (UTM), companies, and the Internet of Medical Things (IoMT). However, as shown in Figure 1, several security threats exist in the cloud [1, 2]. First, cloud service providers cannot be completely trusted. Users think that their data is securely protected if an external cloud is used. However, the service provider may know the data contents stored and utilized on their server. An attacker (a malicious user) can compromise shared data for another security threat. An attacker may access the server, tamper with the stored data, and leak the data. If the data stored on the cloud server is sensitive information, this will pose a significant security threat [3, 4]. Therefore, a security technique that encrypts data stored and transferred in the cloud is required, as is access control for this encrypted data. Of the various security technologies, attribute-based encryption (ABE) ensures secure data encryption/decryption and access control. ABE performs encryption/decryption employing multiple user attributes. It is widely used for secure data sharing in the cloud. ABE schemes include key-policy attribute-based encryption (KP-ABE) and ciphertext-policy attribute-based encryption (CP-ABE). The two ABE schemes depend on the data Access Structure (AS) contained in the ciphertext and the data user secret key. If the AS is included in the ciphertext, the CP-ABE scheme is used, and if the AS is included in the data user secret key, the KP-ABE scheme is

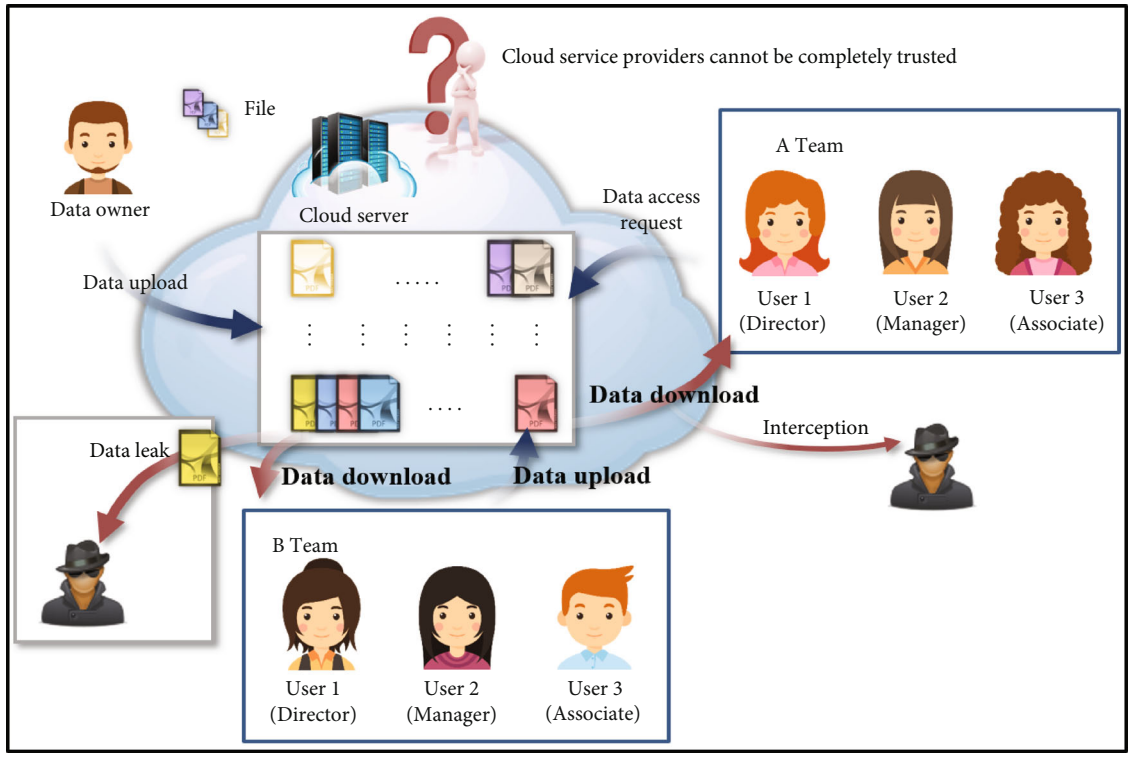


Figure 1: Security threats in the cloud environment.

used. The differences between the two types of ABE schemes are explained in Section 2 [5, 6].

In this paper, we intend to research data sharing in an N:1 cloud environment where data users can decrypt ciphertexts with the attributes of the AS included in the secret key. Here, "N" means multiple users. Since the KP-ABE scheme is suitable in an N:1 cloud environment, research on KP-ABE was conducted. To date, various KP-ABE schemes have been analyzed for secure data storage and sharing technology. However, there are security threats and inefficient schemes among the existing KP-ABE schemes.

First, the traditional KP-ABE schemes encrypt and store data in the cloud that cannot be searched. Therefore, all stored ciphertexts must be decrypted when seeking a desired ciphertext among numerous ciphertexts. This makes the process inefficient. To solve this problem, efforts have been made to introduce searchable encryption [7–10]. However, the number of searches required and the ciphertext size increase proportionally to the number of attributes [11, 12]. This wastes storage space on the server. In addition, when using attribute-based encryption, a server known as the attribute authority (AA) manages user attributes. The AA plays a role in creating secret keys (ciphertext decryption keys) that include public parameters and user attributes. Data owners and users apply the keys to encrypt/decrypt data. If an AA is attacked, users' secret keys may be leaked. Furthermore, most KP-ABE schemes trust their AAs. But still curious AAs can access and decrypt the ciphertexts stored in the cloud with the stored user's attribute information and secret key information. In other words, a key escrow problem may occur by AA [13–15].

In this paper, we propose secure and efficient data storage and sharing system after researching and analyzing ABE to solve the security threats in cloud environments. Our system allows fast ciphertext search, and the ciphertext size is kept constant. The key escrow problem is solved via user key generation. In summary, we establish secure and efficient data storage and sharing system by proposing a searchable key-policy attribute-based encryption (SKP-ABE) system to which various requirements are applied. The contributions of this paper are as follows:

(i) Efficiency of ciphertext search: The cloud server uses searchable encryption technology to quickly search for the ciphertext requested by the user [16, 17]. Compared with existing KP-ABE schemes, this proposed SKP-ABE scheme aggregates the attribute values included in the ciphertext index. In this case, when searching for a ciphertext, it is possible to find the ciphertext in one search regardless of the number of attributes

(ii) Output of ciphertext of constant size: A ciphertext of a constant size is output by aggregating the values of the attributes included in the ciphertext and expressing them as a single value. The size of the ciphertext does not increase according to the number of attributes included in the ciphertext

(iii) Solution of key escrow problem: In existing KP-ABE schemes, the AA generates a key corresponding to the user's AS and transmits it to the user. That is, the AA knows information about the users'

secret keys and attributes. It can sufficiently cause a key escrow problem. In this proposed scheme, the AA creates a partial secret key and sends it to the user. The user creates a final secret key with the received partial secret key that can decrypt the ciphertext. Therefore, the AA does not know the users' secret key information, and the key escrow problem that occurs in an AA is solved

The remainder of this paper is organized as follows: Section 2 describes the research background; ABE is explained. It also describes existing KP-ABE schemes and the KP-ABE security model. Section 3 describes the security requirements to be provided. Section 4 describes the proposed SKP-ABE scheme. Section 5 analyzes the security and efficiency of the scheme, and Section 6 concludes the paper.

2. Background

This section describes ABE and the preliminaries and formulas for understanding it. Then, the KP-ABE system and KP-ABE security model are explained.

2.1. Preliminaries

2.1.1. Bilinear Map. Bilinear mapping has been proposed as a tool to attack elliptic curve cryptosystems in the past. However, recently, it has been used as a cryptography tool for information protection, and the algorithms elliptic curve cryptography (ECC), which are based on bilinear mapping, are widely used in IoT environments. A bilinear pairing function is called a bilinear mapping, and the notation is expressed as follows: Suppose we have multiplicative groups G_1 and G_2 with the same order p. Assume that it is difficult to solve the discrete logarithm problem within a group. Let g be a generator group of G_1, and let $e : G_1 \, X \, G_1 \longrightarrow G_2$ be a bilinear mapping that satisfies the following properties:

(1) Bilinearity: For all $P, Q \in G_1$ and all $, b \in Z_p$, $e(P^a, Q^b) = e(P, Q)^{ab}$

(2) Nondegeneracy: For all $Q \in G_1$, if $e(P, Q) = 1$, then $P = 0$

(3) Computability: For all $P, Q \in G_1$, there is an efficient algorithm to compute $e(P, Q) \in G_2$

2.1.2. Bilinear Diffie Hellman (BDH) Assumption. The deterministic BDH assumption means that, given two pairs $(g^a, g^b, g^c, W = e(g, g)^z)$ and $(g^a, g^b, g^c, T = e(g, g)^{abc})$, there is no algorithm A that can distinguish between the two pairs with meaningful probability. Here, $a, b, c, z \in Z_p$. If algorithm A is able to solve the deterministic BDH assumption, that is $|Pr[A(g^a, g^b, g^c, T) = 1] - Pr[A(g^a, g^b, g^c, W) = 1]| \geq \epsilon$ if satisfied, then algorithm A has an advantage of ϵ [18].

2.1.3. Bilinear Diffie Hellman Exponent (BDHE) Assumption. The deterministic BDHE assumption means that, given $(h, g, g^{\alpha} \cdots ., g^{\alpha\beta}, g^{\alpha\beta+2}, \cdots g^{\alpha 2\beta})$, there is no algorithm A that can compute $T = e(h, g)^{\alpha\beta+1}$ with a meaningful probability. Here, $h, g \in G_1$, $g_i = g^{\alpha^i} (i = 1, \cdots, 2\beta)$ and $g_{\alpha,\beta} = (g_1, \cdots, g_B, g_{B+2}, \cdots, g_{2B})$; when the next two pairs are $(h, g, g_{\alpha,\beta}, W = e(h, g)^z)$, $(h, g, g_{\alpha,\beta}, T = e(h, g)^{\alpha\beta+1})$. If algorithm A is able to solve the deterministic BDHE assumption, that is $|Pr[A(h, g, g_{\alpha,\beta}, T) = 1] - Pr[A(h, g, g_{\alpha,\beta}, W) = 1]| \geq \epsilon$ if satisfied, then algorithm A has an advantage of ϵ [18].

2.1.4. Decisional Bilinear Diffie-Hellman (DBDH) Assumption. Given g^l, g^m, g^n, where $l, m, n \in Z_q$, the DBDH problem is to distinguish g^{lmn} from g^z, where $z \in Z_q$. Given B is an algorithm, and its advantage in solving the problem is $Adv_B^{DBDH}|Pr[A(g^l, g^m, g^n, g^{lmn}) = 1] - Pr[A(g^l, g^m, g^n, g^z) = 1]|$. The DBDH assumption states that the advantage of an algorithm B in solving DBDH problem is negligible.

2.1.5. Elliptic Curve Discrete Logarithm Problem (ECDLP) Assumption. Elliptic curve cryptography can achieve the same security as previous public key encryption methods with fewer bits; it is widely used in IoT and other lightweight environments. Compared to the previous public key encryption methods, it uses short keys, so it is easier to manage the keys, and the encryption is processed at high speed. To use ECC, an elliptic curve is a set of solutions (X, Y) of the equation $y^2 = x^3 + ax + b \pmod p$ defined for arbitrary integers a and b. The fact that the point $P = (X, Y)$ is on the elliptic curve means that the previous equation is satisfied. $Q = x \cdot P$ can be defined for any integer x for two points P and Q. Finding the solution x is the discrete logarithm for elliptic curves. That is, it is easy to find Q by using $x \cdot P$ in Q. However, it is very difficult to infer the value of x even if you know Q and P [19].

2.2. Attribute-Based Encryption

2.2.1. Access Structure. ABE is a scheme of performing encryption/decryption based on an AS created using a set of attributes (e.g., affiliation and occupation) for each entity. Here, the AS is shown in Figure 2. In the access tree, denoted by T, each non-leaf node can represent a threshold gate: an OR gate or an AND gate, depending on the threshold. In general, for all nodes $x \in T$, we use the notations k_x and num_x to represent the threshold of x and the number of children, respectively. For a non-leaf node x, if $k_x = 1$, then x represents an OR gate. If $k_x = num_x$, it represents an AND gate. If $1 < k_x < num_x$, then x is a threshold gate. We define $k_x = 1$ and $num_x = 0$ for leaf node x [5, 6, 20, 21].

2.2.2. Types of ABE. ABE includes CP-ABE or KP-ABE depending on the AS created by the user. In Figure 3(a), the data owner includes the AS when generating the ciphertext and stores it on the cloud server and multiple users can access it. At this time, only if a user's attributes match the attributes of the AS included with the ciphertext can they be decrypted. For example, if the AS is created with {{Director AND Manager} OR Company A}, only users with the Director and Manager attributes among users with the company A attribute can decrypt the ciphertext [5]. The CP-ABE scheme has the advantage of being accessible to any users with the attribute of the AS included in the ciphertext.

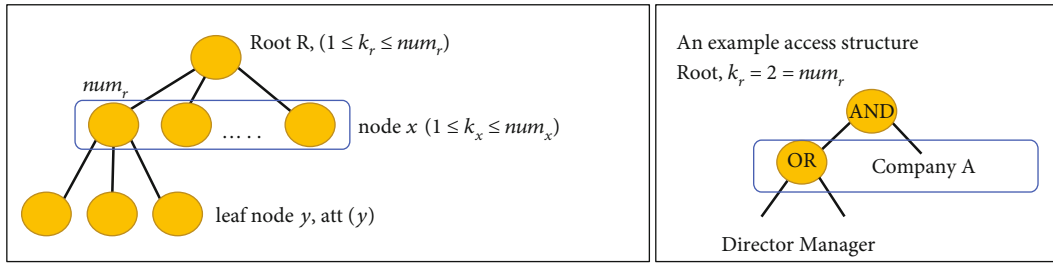


Figure 2: Description of access structure.

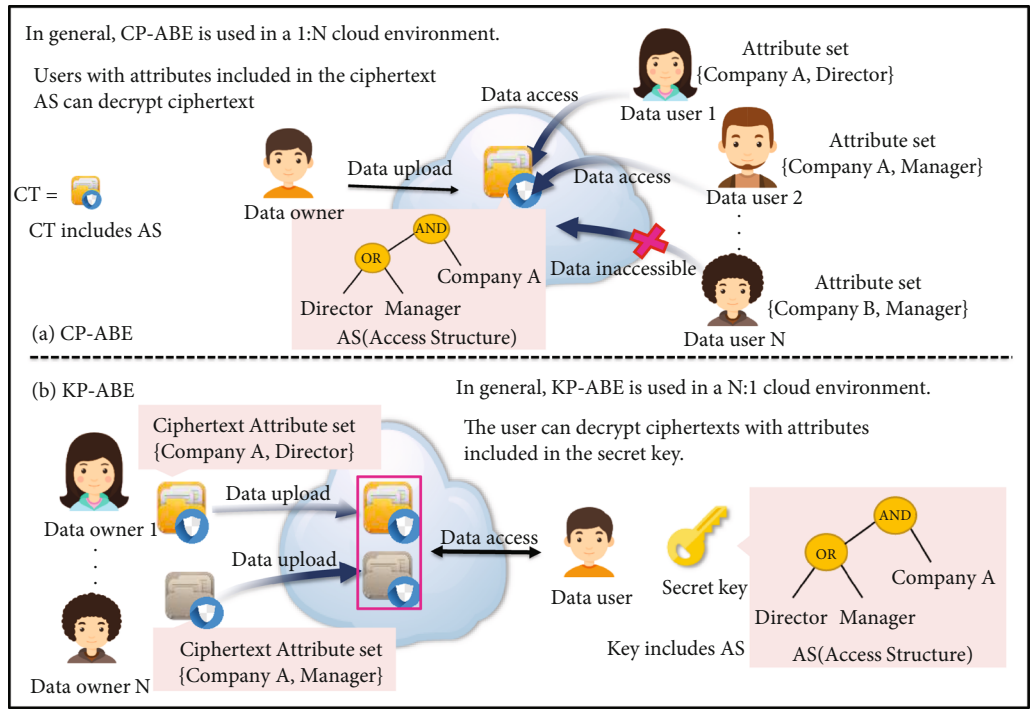


Figure 3: N:N cloud environment using ABE ((a) CP-ABE and (b) KP-ABE).

Therefore, it is widely used in cloud 1:N (N is the number of users) environment. Figure 3(b) shows the KP-ABE scheme. Data users create an AS using their attributes in a KP-ABE scheme and create a corresponding secret, ciphertext decryption key. When data owners generate ciphertexts, they encode the attributes of the users with whom the data will be shared. The ciphertext is stored on the cloud server. Data users can access the cloud server at any later time using a secret key that includes the AS and decrypts the ciphertext with the correct attribute values. For example, if a data owner creates a ciphertext with the attributes {{Director}, {Company A}} and uploads it, only users with the attributes {{Director}, {Company A}} in their AS can decrypt it. In the KP-ABE scheme, when multiple users encrypt data with the attributes of the users who want to share data and upload it to the cloud server, only users with the AS of the attributes designated by the data owner can decrypt the ciphertexts. Therefore, it is widely used in cloud N:1 environment. Figure 3 shows how ABE can be applied to an N: N cloud environment. This paper intends to research a data sharing system in an N:1 cloud environment that allows an authenticated user to decrypt a number of ciphertexts stored with their private key when a large number of data is encrypted and collected and stored. Therefore, research on KP-ABE is suitable.

2.2.3. KP-ABE Model. Figure 4 shows an application of a KP-ABE scheme to cloud environments. There are four entities: an AA, a data owner (users who uploads ciphertext to the cloud), a data user (users who attempts to decrypt ciphertext stored on the cloud), and a cloud storage server. First, a master key and public parameters are generated during the setup phase of the AA. Next, the users create an AS using their attributes, send them to the AA, and request a secret ciphertext decryption key. In a KP-ABE scheme, AS can be created by the user, and an AA can be required to create the AS for the user. In the latter case, the AA generates a secret key corresponding to the user's AS and sends it to the user with the public parameters. When a data owner generates a ciphertext, encryption is performed based on attributes of users that should be allowed access to them. Next, the ciphertext is uploaded and stored on a cloud server. Users registered

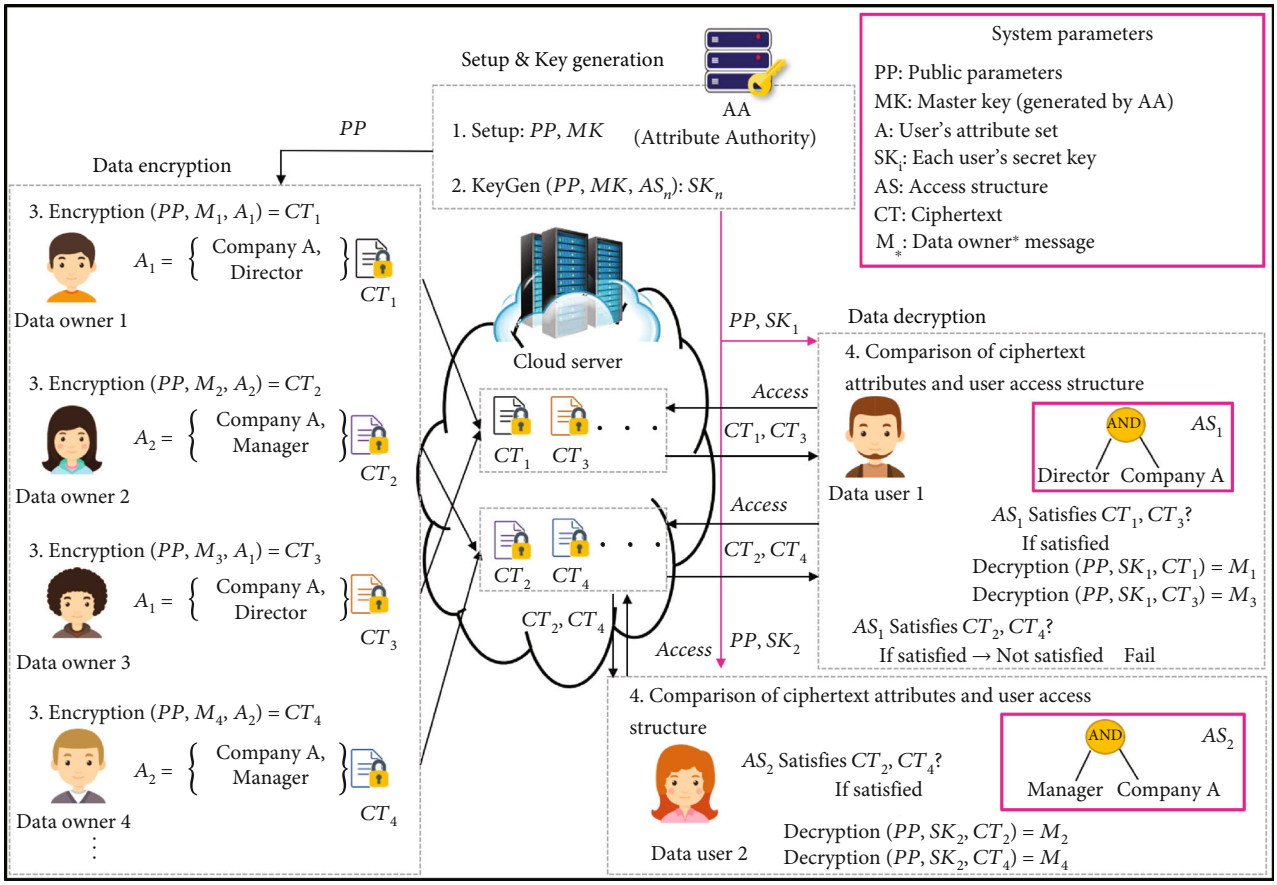


Figure 4: Data sharing scheme using KP-ABE in cloud.

in AA generate tokens and send them to the cloud server. The cloud server transmits the ciphertext requested by the users. Finally, the users obtain the data by decrypting the ciphertext using the AS with that attribute and the received secret key [6, 22].

2.3. Challenges to Build KP-ABE Scheme. Various requirements must be provided to build a secure and efficient data sharing system by applying KP-ABE. The requirements are keyword search, constant-size ciphertext output, key escrow problem solving, verifiable outsourcing, attribute withdrawal, AS anonymization, etc. In order to build a secure KP-ABE scheme, research is needed to provide the above-mentioned requirements. However, the KP-ABE scheme is inefficient because the scheme (model) becomes heavy when all requirements are applied. Therefore, there is a need for research to apply the requirements according to the environment.

The SKP-ABE scheme proposed in this paper is also that provides an existing ciphertext search. The difference from the existing KP-ABE scheme, which provides ciphertext search, is to provide a fast ciphertext search by aggregating the attributes included in the token. In addition, it solved the key escrow problem that occurs in AA and provided a ciphertext of a constant size. Therefore, it provides better requirements than the KP-ABE scheme, which provides only the existing keyword search.

2.3.1. Searchable Encryption. As cloud computing develops, users store and manage large amounts of data using storage space provided by an external service provider such as Google cloud. However, when sensitive personal information is stored externally, security issues arise. Therefore, it is important to encrypt all data. However, then the cloud server must decrypt all stored ciphertexts to find data requested by a user. This is very computationally inefficient [7–10]. One of the security technologies to solve this is searchable encryption. Data can be found without decrypting the ciphertext requested by the user. Therefore, when multiple owners encrypt and store data on the cloud, users can efficiently locate the desired ciphertext.

An early version of searchable encryption, proposed by Song, Wagner, and Perrig in 2000, is a hidden search designed to be searchable without leaking plaintext information [23]. However, the initial version lacked a clear definition of security. Since then, searchable encryption systems that use symmetric or asymmetric keys have attracted much attention. Currently, searchable encryption technology is used with ABE to improve ciphertext search efficiency [7–10].

Figure 5 shows a KP-ABE scheme with searchable encryption applied. The existing KP-ABE scheme assumes that when a user requests a ciphertext from the cloud server, the cloud server transmits the ciphertext to the user. However, KP-ABE schemes with searchable encryption add the phase of searching for a ciphertext on the cloud server.

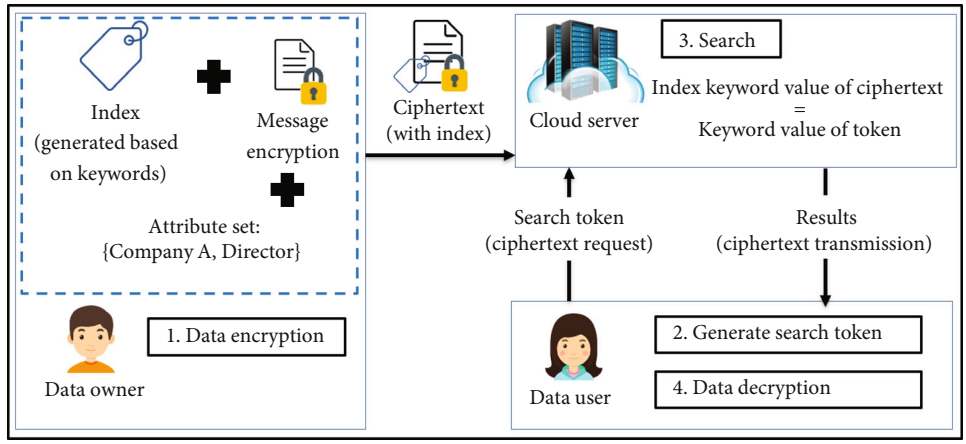


Figure 5: KP-ABE-based data sharing model applying searchable encryption in cloud environment.

In detail, the ciphertext is retrieved from the cloud server based on keywords and attribute values. The data owner selects keywords and attribute values, creates an index, and uploads it to the cloud along with the ciphertext. Next, the user creates a search token using keywords and attribute values to find the ciphertext. Then, it is sent to the cloud server to request the ciphertext. The cloud server searches for the ciphertext by comparing the stored ciphertext index value (including keyword values and attributes) with the search token values (including keyword values and attributes). If matching ciphertexts are found, they are sent to the user. The cloud server finds the requested ciphertext but does not decrypt it [24].

2.3.2. Key Escrow Problem. Key escrow is a system that entrusts encryption (secret) keys to a third party (server) and stores them. If the user key is damaged or lost, the previously entrusted secret key can be issued through the server. However, a server that knows the information about the key may cause a key escrow problem that may attempt to access and decrypt the ciphertext. As a result, user data may be leaked, and various security threats such as abuse of access rights may occur. From the past to the present, in various cryptographic research fields such as key recovery, signature, and ABE, it often occurs in servers (key generation center (KGC) and AA, etc.) that generate and manage keys [25–28]. In an environment where a key escrow problem occurs, it is assumed that users do not completely trust the server managing the key. Therefore, entrusting all key information to the server is a risk factor [29].

The AA is a trusted server that manages properties and generates keys in a data sharing environment using KP-ABE. However, in some KP-ABE schemes, AA is recognized as a semi-trusted server that manages user attributes, so it is mentioned that key escrow problems can occur sufficiently in AA. The term semi-trusted means that the AA is not fully trusted because it has information about the users' secret keys that could cause a key escrow problem. The AAs are honest but curious and have the right to view user information at any time. In the KP-ABE scheme, AA generates a ciphertext decryption key corresponding to the user's attributes and transmits it to the user. Since the AA knows your secret key, it can use it to access the cloud and crack your ciphertext. Therefore, research is being conducted from the existing KP-ABE scheme with single AA to the KP-ABE scheme with multi-AA scheme. This research aims to prevent a key escrow problem in advance with the users' key and attribute information that the AA alone knows [13, 28].

In the multi-AA scheme, when a user requests a secret key by global identity (GID), values corresponding to user attributes are calculated in each AA to create a secret key and send it to the user. Although there is a scheme in which the user generates a secret key with the attribute value received from the AA, usually, multi-AAs generate a secret user key and send it to the user. Above all, since multi-AAs share information about the users' secret key, the AA cannot independently cause a key escrow problem. However, the multi-AA scheme has a disadvantage. The amount of computation required to generate a user secret key increases according to the number of AAs, and a collusion attack between AAs must also be considered. Furthermore, in some KP-ABE schemes, the multi-AA scheme is also viewed as a concept managed by a Central Authority.

2.4. Related Work. In 2006, an initial version of the KP-ABE system was proposed, and based on this, research was conducted to satisfy various requirements. This SKP-ABE scheme provides ciphertext search, constant-size ciphertext, and key escrow problem solving. Table 1 lists an analysis of existing KP-ABE schemes. The description of the KP-ABE scheme that provides the ciphertext search is as follows.

Yin et al. [7] developed a model that adds searchable encryption to the KP-ABE scheme. It is useful when searching for ciphertext in a cloud that manages big data, but the ciphertext size increases with the number of attributes. In addition, as the data owner creates a secret key and transmits it to the user via a secure channel, the data owner knows its secure key. Thus, a key escrow problem may occur.

Ameri et al. [8] considered an environment where the cloud provider was not completely trusted. Their scheme allows the creation of a search token at any time. This token matches all ciphertexts containing the keyword. However, as information leakage is possible, Ameri et al. proposed KP-ABE schemes, in which the search token matches only

Table 1: Comparison of KP-ABE schemes.

KP-ABE scheme	Ciphertext search	Key escrow problem	Ciphertext size
Yin et al. scheme [7]	Provided	Possible to occur	Proportional to the number of attributes
Ameri et al. scheme [8]		Not considered (possible to occur)	
Li et al. scheme [9]			
Meng et al. scheme [10]			
Longo et al. scheme [13]	Not provided	Key escrow problem solved (multi authority)	Constant-size ciphertext
Leyou Zhang et al. scheme [14]		Key escrow problem solved (decentralized authority)	Proportional to the number of attributes
Kai Zhang et al. scheme [30]		Not considered (possible to occur)	Constant-size ciphertext
Belguith et al. scheme [31]			
Goal of the proposal scheme	Provided (fast ciphertext search)	Key escrow problem solved (single authority)	Constant-size ciphertext

ciphertext generated within a specified time interval [8]. That is, it is a scheme that can share ciphertexts within a specified time frame using temporary keywords. Nonetheless, they did not consider the key escrow problem. They assumed that the AA was fully trusted. However, since the AA knows the users' key information, this can cause a key escrow problem. Also, ciphertext size increases by the number of attributes included in the ciphertext.

Li et al. [9] proposed a secured ABE scheme with a searchable encryption function to protect the security and privacy of sensitive data. To counter keyword-guessing attacks, all keywords were signed using secret keys of the data owners when generating ciphertexts. However, depending on the number of attributes, it can increase the size of the ciphertext, and it has the key escrow problem.

Meng et al. proposed a scheme that improved computation efficiency by using a constant-size output ciphertext and a constant pairing operation in a KP-ABE scheme that provides searchable encryption [10]. However, the key escrow problem remained possible.

Figure 6 shows how ciphertext is searched on the cloud server. It assumes that three ciphertexts are stored on the cloud server, each with two attributes. When the server searches for a ciphertext, the first search compares the first attribute of the token with the first attribute of the ciphertext. The second search compares the second attribute of the token with the second attribute of the ciphertext and finds a matching ciphertext. In Figure 6(a), the number of ciphertext searches increases proportionally to the number of attributes contained in the token and ciphertexts. For example, the searchable KP-ABE scheme was mentioned above (Yin et al., Ameri et al., Li et al., and Meng et al.). To solve this problem, an aggregate operation is performed on the attribute value included in the ciphertext and the attribute value of the token generated by the user. Then, the aggregated attribute values of the token and the ciphertext are compared to find a matching ciphertext [16, 17, 32]. Figure 6(b) shows the aggregate attributes of tokens and ciphertexts when searching for a ciphertext. As a result, the number of ciphertext searches is not affected by the number of attributes contained in the tokens and ciphertexts. The disadvantage is that tokens can be generated in multiple ways depending on the aggregate attributes of the ciphertext that the user wants to find. However, if an aggregation operation is used, searching for a ciphertext requested by the user on the server will be more efficient than the scheme in Figure 6(a). In terms of decryption, since the goal is to find the ciphertext in most KP-ABE schemes that provide searchable encryption, the decryption process of the ciphertext is omitted. Therefore, partial decryption is not provided.

The KP-ABE scheme that solves the key escrow problem and constant size is as follows. The KP-ABE schemes of Longo et al. [13] and Leyou Zhang et al. [14] solved the key escrow problem using a multi-AA or decentralized AA. By dividing the key generation authority of a single AA among multiple AAs, no individual AA knows all of the information about a users' secret key. However, the ciphertext search function is not provided, and constant-size ciphertext and partial decryption are provided depending on the scheme. The schemes of both Kai Zhang et al. [30] and Belguith et al. [31] output constant-size ciphertext [26, 27].

2.5. KP-ABE Security Model Definition. The security goal of searchable ABE is to prevent an attacker from obtaining information about a keyword from the search token and index keywords in the ciphertext. In other words, if a search token is not found, it should not disclose information about the index keyword w. KP-ABE schemes must provide security against attackers who can obtain search tokens for arbitrary keywords w of their choosing. Even in these attacks, the attacker should not be able to distinguish the encryption of the keyword w_1 and the encryption of the keyword w_0, which does not include obtaining the trapdoor [7, 16]. We use an adaptive chosen keyword attack game and an adaptive chosen plaintext attack game to define the security model of search tokens and index keywords. We provide a formal definition of security through the following games between a probabilistic polynomial-time attacker A and challenger C.

2.5.1. Adaptive Chosen Keyword Attack Game

(1) Challenger C executes Setup(1^λ) to generate master key MK and public parameter PP. Then, it sends the PP to attacker A

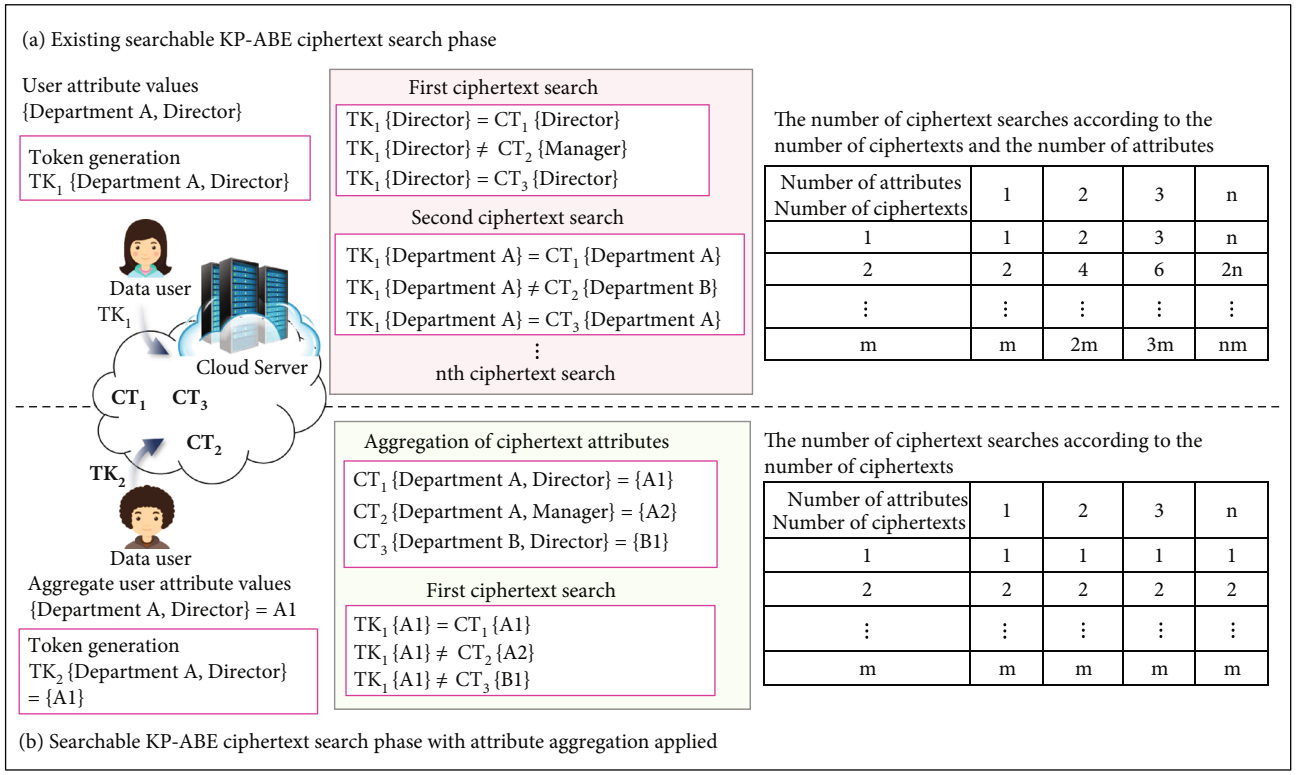


Figure 6: Existing searchable KP-ABE ciphertext search phase.

(2) Attacker A can adaptively query the ciphertext for all search keywords. Accordingly, when A requests the ciphertext for the search keyword w_b, C generates the ciphertext as Index(w_b) and sends it to the attacker A

(3) Attacker A selects two keywords w_0 and w_1 and sends them to challenger C. C fairly selects a random bit value as $b \in \{0, 1\}$, has the attribute set $\{s_1 \ldots s_n\}$ received from the attacker, and encrypts it with w_b to generate Index(w_b). And it sends the ciphertext index to the attacker

(4) Attacker A continuously requests a private key query from challenger C and generates a legitimate search token by encrypting the query keyword w (w is expressed as w_0 or w_1)

(5) Attacker A guesses that b is b'. We define the advantage that attacker A wins in the above game within stochastic polynomial time as $|\Pr[b = b'] - 1/2|$.

Definition 1. Searchable ABE is semantically secure against adaptive chosen keyword attacks in the above security game when the attacker has at most a negligible advantage in probabilistic polynomial time (PPT). That is, in the chosen keyword attack model, the search token and index keyword should not expose the plaintext information of the query keyword.

2.5.2. Adaptive Chosen Plaintext Attack Game

(1) Challenger C executes Setup(1^λ) to generate master key MK and public parameter PP and sends PP to attacker A. A sends a set of attributes $\{s_1, \ldots, s_n\}$ that it wants to test to C

(2) Attacker A requests a secret key query corresponding to the access structure $\{AS_1, \ldots, AS_n\}$from C. At this time, the limitation is that the set of attributes $\{s_1, \ldots, s_n\}$ must not satisfy the access structure $\{AS_1, \ldots, AS_n\}$. Attacker A receives the secret key from C, encrypts the keyword to be queried, and generates a search token

(3) Attacker A selects two messages M_0 and M_1 and sends them to challenger C. C fairly selects a random bit value as $b \in \{0, 1\}$, and encrypts it as CT(M_b) with the attribute set $\{s_1, \ldots, s_n\}$ received from the attacker. And it sends the ciphertext CT(M_b) to the attacker

(4) Attacker A continuously requests the secret key query corresponding to the access structure $\{AS_1, \ldots, AS_n\}$ from challenger C as in (2). Restrictions here are the same as in (2).

(5) Attacker A guesses that b is b'. We define the advantage that attacker A wins in the above game within stochastic polynomial-time as $|\Pr[b = b'] - 1/2|$.

Definition 2. Searchable ABE is semantically secure against adaptive chosen plaintext attacks in the security game above if the attacker has at most a negligible advantage in PPT.

3. Security Requirements

This section describes the requirements in terms of security and efficiency, such as data encryption/decryption and data access for secure and efficient data storage and sharing in the cloud.

(i) Shared data confidentiality and integrity: If data stored and shared in the cloud is in plain text, the data is exposed to various security threats. Therefore, security for the shared data is required, and the confidentiality and integrity of shared data must be ensured. The ciphertext should be decryptable only by legitimate users

(ii) No access for unauthorized users: If anyone can access cloud data, various security threats arise. Thus, access control is required. ABE is a security and access control technology. Only an authenticated user can decrypt accessed data by comparing an attribute value specified by the data owner with the AS attribute value of the user's secure key. Thus, users without the correct attributes cannot decrypt the data even if they access it

(iii) Ciphertext search efficiency: It is difficult for a user to search for the desired data among the numerous ciphertexts stored in the cloud. To search for a ciphertext requested by a user, all stored ciphertexts must be decrypted to check the contents of their data. This is inefficient. Therefore, searchable encryption technology which enables users to search for the requested data without decryption is essential [7–10]. However, in some of the existing schemes, the number of searches increases proportionally to the number of attributes when searching for a ciphertext. Therefore, in the KP-ABE scheme, it is necessary to aggregate the values of the attributes corresponding to the ciphertext keywords. As a result, the user should quickly search for the desired ciphertext

(iv) Constant-size ciphertext: In existing KP-ABE schemes, the size of the generated ciphertext is proportional to the number of included attributes. Cloud storage space is used inefficiently due to the increased ciphertext size [11, 12]. Therefore, it is needed to research in which the size of the ciphertext can be constant output regardless of the number of attributes

(v) The key escrow problem: Since the AA knows information about the users' secret keys, it is cannot be fully trusted because that can cause a key escrow problem. Therefore, it is necessary to reduce AAs secret key generation authority. Specifically, the key escrow problem can be solved by generating a secret key using multiple AAs. For example, a user receives a partial secret key from the AA and generates the final secret key [33, 34].

4. The Proposed SKP-ABE Scheme

In this section, our proposed SKP-ABE scheme is described (see Figure 7). When searching for a ciphertext, the attribute values of the token and ciphertext are aggregated and compared.

Therefore, it is possible to find the requested ciphertext quickly. Furthermore, the key escrow problem on an AA is solved by generating a final ciphertext decryption key using a partial secret key received from the AA. In addition, by using a constant-size ciphertext, the effects of attribute number on ciphertext size are minimized. Finally, the cloud server finds the ciphertext and sends it to the user, and the user decrypts it to obtain data.

4.1. System Model

4.1.1. System Entities

(i) Data Owner: The data owner encrypts data and uploads it to the cloud. The owner generates a ciphertext with the attributes of the users who can access the data. Then, an index is created by selecting keywords that can represent the ciphertext (CT). Finally, the CT and index are uploaded together to the cloud server

(ii) Cloud Server: In general, a cloud server includes a storage server in which data is stored and a server that performs operations. For example, the cloud server stores and manages data. When a user requests ciphertext, the server performs a ciphertexts search using the ciphertext index and token value received from the user. After that, the retrieved ciphertexts are sent to the user

(iii) Attribute Authority: The AAs are honest but curious and have the right to view user information at any time. In this proposed SKP-ABE scheme, the secret key generation phase of the AA is modified to the partial secret key generation phase. In addition, when registering a user, a certificate that can be authenticated is generated and then sent to the user. The certificate is used to verify that the user is registered when the user later accesses the cloud server

(iv) Data User: A data user is an entity that downloads and decrypts ciphertext uploaded to the cloud. The user generates a final secret key (FSK) to decrypt the ciphertext using the PSK received from the AA. In addition, by selecting the keywords of the ciphertext to be found, a token is generated. A user can request ciphertexts from the cloud server with a token. When the user receives the ciphertext from the cloud server, it uses FSK to perform decryption to obtain the ciphertext to obtain the data

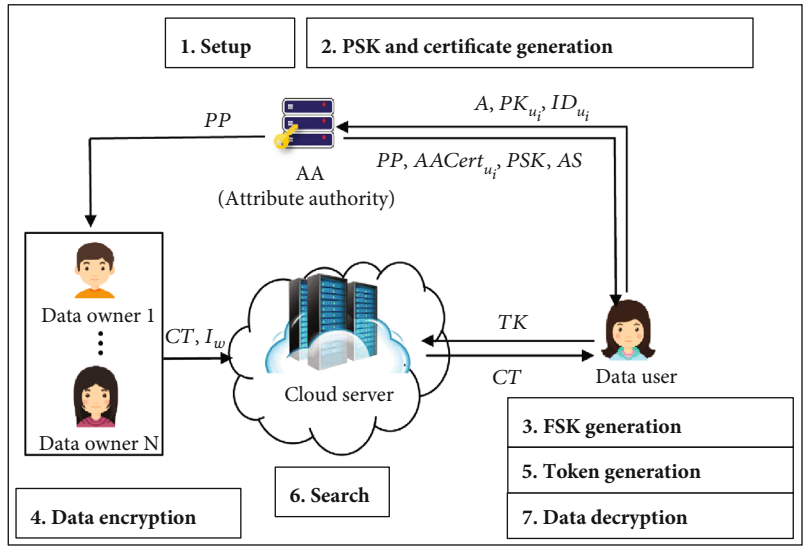


Figure 7: This proposed SKP-ABE scheme scenario

Table 2: Notations.

Symbol	Definition
p, q	Prime order
PP, MK, PK_{AA}	Public parameters, master key, AA's public key
PK_{u_i}, SK_{u_i}	Data user public/private key pair
PSK	Data user partial secret key (partial decryption key)
FSK	Data user final secret key (ciphertext decryption key)
ID_{u_i}	User identifier
$AACert_{u_i}$	Data user certificate
A_U, A	User attribute data, A set of attribute data
AS	Access policy or access structure
$T_{w'}$	Token with keyword w′ (ciphertext search token)
w, w′	Keywords for data owners, keywords for data users
I_w	The ciphertext index value generated based on the keywords
CT	Ciphertext
$H_1(\bullet)$	Cryptographic hash function $\left(\{0,1\}^* \longrightarrow Z_p^*\right)$
$H_2(\bullet)$	Cryptographic hash function $\left(\{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \longrightarrow Z_p^*\right)$

4.1.2. System Parameters. The system parameters used in the proposed SKP-ABE scheme is shown in Table 2.

4.1.3. Procedure. This proposed SKP-ABE scheme provides secure and efficient data storage and sharing in cloud environments. Compared to existing KP-ABE schemes shown in Table 1, the proposed SKP-ABE scheme meets more requirements. This SKP-ABE scheme consists of 7 phases. The phase are as follows.

(i) Setup(k): The AA generates master key (MK) and public parameters (PP) with security parameter k as input. The data user generates a private/public key pair

(ii) PSK and CertGen(MK, A, PP, ID_{u_i}, PK_{u_i}) $\longrightarrow$ $AACert_{u_i}$, PSK, AS: When a user requests registration and a partial secret key from the AA, the AA creates AS based on the user's attributes. Next, it generates a partial secret key (PSK) also based on the user's attributes, creates a certificate ($AACert_{u_i}$) based on the user's ID_{u_i} and public key, and sends them all to the user

(iii) FSKGen(PSK, AS) $\longrightarrow$ FSK: The user receives PSK and AS from the AA and generates the final secret key (FSK) corresponding to the AS

(iv) Encrypt(PP, M, S, w) $\longrightarrow$ CT, I_w: The data owner selects the message (M) and encrypts with the attribute sets (A) and PP of the users who can access their data. In addition, index value (I_w) is created by selecting keywords that represent the CT and transmitted to the cloud server along with the CT. A keyword is a word that can represent a CT and is known only to the data owner and user

(v) TokGen(FSK, w′) $\longrightarrow$ TK: The user generates a token $T_{w'}$ to find the CT in the cloud. At this time, a token is generated with the keywords w′ of the CTs to be found and the FSK received from the AA. It then signs the token with the certificate and sends the TK to the cloud to request the CT

(vi) Search(I_w, $T_{w'}$) $\longrightarrow$ {0, 1}: The cloud server verifies that the CT of the registered user is requested through $AACert_{us_i}$. Then, the CT requested by the user is found using the received $T_{w'}$ and the CT index (I_w). The searched CT is expressed as {0, 1},

and as a result, 0 means not found, and 1 means found. The retrieved ciphertexts are sent to the user

(vii) Decrypt(CT, AS, l, FSK, PP) $\longrightarrow$ M: When the user receives the CTs from the cloud server, it decrypts by comparing the attribute value in the AS with the attribute value in the CT. If the decryption is successful, the user can obtain a message M

4.2. Description of the Proposed SKP-ABE Scheme. The AA generates two cycle multiplication groups G and G_T of prime order p and generates a bilinear map $e : G \times G \longrightarrow G_T$ ($e : G \times G \longrightarrow G_T, \forall i, j \in G, e(i, j) = v, v \in G_T$). Let g denote a generator of G. The AA generates a subgroup G_2 of elliptic curve points of prime order q and chooses a generator P of G_2. Elliptic curve point-based crypto-operations are used to generate user keys, and key security assumes the intractability of ECDLP. Here, the user key means the initially generated key pair (PK_{u_i}, SK_{u_i}) for users to register with AA. Assume that there are n attributes in the universe where the universal set is $A = \{Att_1, Att_2, Att_3, \cdots, Att_n\}$. $W = \{W_1, W_2, W_3, \cdots, W_n\}$ is an AS and includes attributes, such as $Att_i \subset W_i$.

4.2.1. Setup Phase. Initially, the AA creates PP, MK, and PK_{AA} in the setup phase. The AA generates random values $\alpha, k \in Z_p^*$, $t_i \in G$ and computes $f = g^k$, $EP = e(g, g)^{\alpha}$.

The public parameters, master key, and public key are generated as follows:

$$PP = <G, G_T, G_2\ e, g, \left\{T_i = g^{t_i}\right\}_{i\in[1,n]}, f = g^k, EP = e(g, g)^{\alpha}, H_1, H_2 >, \quad (1)$$

$$MK = <\alpha, \{t_i\}_{i\in[1,n]} >, PK_{AA} = <\alpha \bullet P > . \quad (2)$$

The data user selects a random value for $x_{u_i} \in Z_p^*$ and generates a private key/public key pair as follows:

$$PK_{u_i}, SK_{u_i} = <x_{u_i} \bullet P, x_{u_i} > . \quad (3)$$

The user requests registration and a partial secret key by transmitting their attribute set $A = \{Att_1, Att_2, Att_3, \cdots, Att_n\}$, public key (PK_{u_i}), and identifier (ID_{u_i}) to the AA.

4.2.2. PSK and Certificate Generation Phase. The AA creates an access tree AS with a leaf node l value set based on user attributes. And PSK is created with the attribute value corresponding to AS. In addition, the user's public key and ID are used to generate a certificate. The AA sends the PP to the data owner and PP, $AACert_{u_i}$, PSK, and AS to the user:

$$D_{i,j} = g^{t_n A} \text{or } g^{t_{n+1} A}, H_{Att_i} = H_1(Att_i)^k{}_{i\in[1,n]}, \quad (4)$$

$$PSK = <g^{\alpha}, \left\{D_{i,1}\right\}_{i\in[1,n]}, H_{Att_i} > . \quad (5)$$

When creating a certificate, select $o_{u_i} \in Z_p^*$, $O_{u_i} = o_{u_i} \bullet P$.

$$d_{u_i} = o_{u_i} + \alpha H_2\left(ID_{u_i}, PK_{u_i}\right), AACert_{u_i} = \left(O_{u_i}, d_{u_i}\right). \quad (6)$$

4.2.3. FSK Generation Phase. Then, the user selects a random value and generates an FSK with the PSK and AS received from AA:

Random number $r_i \in Z_p^*$, $r = \sum_{i=1}^{n} r_i$.

$$D_i = g^{\alpha+r}, D_i' = g^{r_i}, \quad (7)$$

$$FSK = <AS, D_i, D_i', \left\{D_{i,1}\right\}_{i\in[1,n]}, H_{Att_i} > \quad (8)$$

4.2.4. Data Encryption Phase. The data owner creates a ciphertext with the PP and the attribute of the user that can access the data. Then, keywords representing the ciphertext are selected, and an index value I_w is generated for the keyword and transmitted together with the CT (see Equations (9)-(12)).

Select message M and add random numbers $s_i, s' \in Z_p^*$, such that $s = \sum_{i=1}^{n} s_i$.

Select attribute set $A = \{Att_1, Att_2, Att_3, \cdots, Att_n\}$ and keyword w (the keyword is a value that indicates the ciphertext created by the data owner and requested by the user. A ciphertext index can use a single keyword, and multiple keywords are more secure).

$$C_0 = M \bullet EP^s, C_1 = h^s, \quad (9)$$

$$C_2 = g^s \bullet \prod_{i\in n} g^{t_i Att_i}, C_3 = \prod_{i=1}^{n} H_1(Att_i)^s, \quad (10)$$

$$\tilde{C}_1 = e\left(f, g^{ws'}\right), \tilde{C}_2 = g^{ss'}, \quad (11)$$

$$CT = <A, C_0, C_1, C_2, C_3 >, I_w = <\tilde{C}_1, \tilde{C}_2, C_3 > . \quad (12)$$

An index value of I_w is set for each CT. The data owner sends CT and I_w to the cloud server. The cloud server securely stores the CT and I_w received from the data owner.

4.2.5. Token Generation Phase. The user selects a keyword w' in the ciphertext to found. Then, the user generates a token $T_{w'}$ using FSK that can be used to find a ciphertext. After token generation, the token is signed with the certificate received from AA (see Equations (13) and (14)).

Select a keyword to search for and generate a token.

$$T_{w'} = e\left(\prod_{i=1}^{n} H_{Att_i}, g^{w'}\right). \quad (13)$$

Sign using a certificate:

$$AACert_{us_i} = d_{u_i} + x_{u_i} H_2\left(ID_{u_i}, PK_{u_i}, T_{w'}\right). \quad (14)$$

The user requests a ciphertext by sending $TK = (T_{w'}, AACert_{us_i})$ to the cloud server.

4.2.6. Search Phase. The cloud server verifies the registered user *i* and token through $AACert_{us_i}$. After verification, the $T_{w'}$ received from the user is compared to the I_w of the CTs stored on the server, and matching ciphertexts are

found. This will only happen when the keyword w' selected by the user and the keyword w selected by the data owner are the same. The search result is displayed as $\{0, 1\}$. The retrieved ciphertexts are sent to the user:

$$\begin{aligned} \mathrm{AACert}_{us_i} \bullet P &= o_{u_i} \bullet P + \alpha \bullet P * H_2\left(ID_{u_i}, PK_{u_i}\right) \\ &\quad + x_{u_i} \bullet P * H_2\left(ID_{u_i}, PK_{u_i}, T_{w'}\right) \\ &= O_{u_i} + PK_{AA} * H_2\left(ID_{u_i}, PK_{u_i}\right) \\ &\quad + PK_{u_i} * H_2\left(ID_{u_i}, PK_{u_i}, T_{w'}\right), \end{aligned} \tag{15}$$

Ciphertext search: $e(\tilde{C}_2, T_{w'}) = e(\tilde{C}_1, C_3)$

$$\begin{aligned} e\left(\tilde{C}_2, T_{w'}\right) &= e\left(g^{ss'}, e\left(\prod_{i=1}^{n} H_{Att_i}, g^{w'}\right)\right) \\ &= e\left(e\left(g^{s'w}, g^{k}\right), \prod_{i=1}^{n} H_1(Att_i)^s\right) = e\left(\tilde{C}_1, C_3\right). \end{aligned} \tag{16}$$

4.2.7. Data Decryption Phase. The user performs decryption by comparing the attribute value specified in the user AS with the attribute values included in the ciphertext. Parameter l refers to the attribute value (leaf-node) of the user AS. If the decryption is successful, the users obtain M (see Equations (17) and (18)).

Access structure $W = \{W_1, W_2, W_3, \cdots, W_n\}$.

If $Att_i \in W_i$, compute $D_{i,j} = (g^{t_n})^{l_n}$.

If $Att_i \notin W_i$, compute $D_{i,j} = (g^{t_{n+1}})^{l_n}$.

$$\begin{aligned} C &= \frac{e\left(C_1, \prod_{j\in A} D_{i,j}\right)}{e\left(C_2, f \bullet \prod_{j\in A} D_i'\right)} = \frac{e\left(C_1, \prod_{j\in A} (g^{t_n})^{l_n}\right)}{e\left(C_2, \left(g^{k} \bullet \prod_{j\in A} g^{r_i}\right)\right)} \\ &= \frac{e\left(g^{ks}, \prod_{j\in A} (g^{t_n})^{l_n}\right)}{e\left(\left(g^{s} \bullet \prod_{j\in A} (g^{t_n})^{l_n}\right), \left(g^{k} \bullet \prod_{j\in A} g^{r_i}\right)\right)}, \\ &\frac{e\left(g^{ks}, g^{\sum_{i=1}^{n} t_n l_n}\right)}{e\left(\left(g^{s}, g^{\sum_{i=1}^{n} t_n l_n}\right), \left(g^{k}, g^{r}\right)\right)} = \frac{e(g,g)^{ks}}{e(g,g)^{ks+sr}} = e(g,g)^{-sr}, \end{aligned} \tag{17}$$

$$\begin{aligned} M &= \frac{C_0}{e(C_1, D_i) \bullet C} = \frac{M \bullet EP^s}{e(g^s, g^{\alpha+r}) \bullet e(g,g)^{-sr}} \\ &= \frac{M \bullet e(g,g)^{\alpha s}}{e(g,g)^{\alpha s + rs - rs}}. \end{aligned} \tag{18}$$

5. Analysis of Proposed SKP-ABE Scheme

This proposed SKP-ABE scheme was analyzed for security and efficiency to satisfy the security requirements detailed in Section 3. Table 3 is an analysis table comparing the existing scheme and the proposed SKP-ABE scheme in terms of security and efficiency.

5.1. Security Analysis

(i) Shared data confidentiality and integrity: Data confidentiality and integrity are protected because data are encrypted, stored, and shared using a KP-ABE scheme. Data is encrypted using attributes $A = \{Att_1, Att_2, Att_3, \cdots, Att_n\}$. Therefore, only a user with an AS matches the attributes for the ciphertext and has the corresponding FSK can decrypt and obtain the data. An attacker who steals data cannot decrypt it

(ii) Access control: In the existing KP-ABE scheme, if the user had the secret key received from AA, the user could create a token and request a ciphertext by accessing the cloud. It is possible to access the cloud and request a ciphertext without further authentication. Ciphertext is decrypted using the user attributes. However, if anyone can access the cloud server, it is difficult to restrict the users. If anyone can access the cloud server, it is difficult to restrict the users. Furthermore, data theft or forgery may occur if a user is malicious. Therefore, an access control function that ensures that only registered users can access the cloud server is required. In the proposed SKP-ABE scheme, only users registered by the AA can access the cloud and request a ciphertext. Each registered user receives an $AACert_{u_i}$ from the AA. The cloud server verifies the validity of $AACert_{us_i} \bullet P = O_{u_i} + PK_{AA} * H_2(ID_{u_i}, PK_{u_i}, T_{w'})$ using the user's public key PK_{u_i} and ID_{u_i}, $T_{w'}$ and O_{u_i}; the ciphertext search is performed. Then, the found ciphertext is sent to the user. Therefore, unauthorized users or third parties cannot access the cloud server other than registered users

(iii) Key escrow problem: To solve the key escrow problem, the AA does not know the users' secret key information completely. In our proposed SKP-ABE system, the user receives a partial secret key from AA and generates a final secret key. The value $(D_i = g^{\alpha+r}, D_i' = g^{r_i})$ included in the final secret key is a value required for the users to decrypt data, and only the user who generated the final secret key knows. In the SKP-ABE system, when requesting a ciphertext, an access token signed with a certificate is required, so AA cannot generate it and therefore cannot request a ciphertext. If it is assumed that the AA acquires the user's partial secret key and search token and accesses the cloud, it can search for and attempt to decrypt the ciphertext but cannot finally decrypt the ciphertext. This scheme is similar to solving the key escrow problem from KGC in the certificate-based signature. It was applied to our proposed scheme. In the phase where AA issues $ACert_{u_i}, PSK, AS$ to the user, even if the attacker obtains $AACert_{u_i}, PSK, AS$, the attacker cannot access the cloud with the obtained certificate because he does not know the users' private key. In general, an ABE scheme assumes that the

Table 3: Comparison between the Proposed SKP-ABE scheme and the existing KP-ABE scheme.

	Yin et al. scheme [7]	Ameri et al. scheme [8]	Li et al. scheme [9]	Longo et al. scheme [13]	Zhang et al. scheme [14]	Proposed scheme
Ciphertext search	Provided			Not provided		Provided
Number of searches	Number of searches increases by the number of attributes					Only one search is required because of attribute aggregation (fast search)
Key escrow problem	Key escrow issues not considered (key escrow problems may occur enough)			Key escrow problem solved (multiauthority)		Key escrow problem solved (user generates decryption key)
Ciphertext size	Proportional to the number of attributes			Constant-size ciphertext	Proportional to the number of attributes	Constant-size ciphertext
Secret key (ciphertext decryption)	$3nT_E$	$3nT_E + nH$	$(4n+1)T_E + 2nT_M$	$K((n+1)T_E + 3nT_M)$	$(2n+2K+5)T_E + 2T_M$	$(2n+2)T_E + nH + (n-1)T_M + E$
Encryption (ciphertext)	—			$(nK+1)T_M$ + $(n+1)T_E$	$(2K+n+2)T_E$ + 2 $(K-1)T_M + kP + M$	$(n+3)T_E + nT_M + M$
Encryption (ciphertext index)	$(n+1)T_E$ + $E + nH$	$(n+4)T_E$ + $(n+2)H + 5M$	$(n+6)T_E + T_M + 2H$	Not provided(this is index encryption search)		$P + (n+4)T_E + (n-1)T_M + nH$
Search (test)	$2nP + nE$	$(2n+1)P + lE$	$(2n+1)P + nT_E + nT_M$			$3P + (n-1)T_M + nT_E$
Decryption (user)	—			$nP + nT_E$+(n-1) T_M	$(K+1+n)P + nE$	$3P + (2n+1)T_M + nT_E$

P: pairing operation; M: multiplication operation; E: exponentiation operation; n: number of attributes; H: hash function; T_E : *Exponentiation* in G; T_M : *Multiplication* in G; K : *Number* of *Attribute Authority*.

communication channel between the AA and the data owner and between the AA and the data user is a secure

(iv) Protection against chosen keyword attacks using a secure game model: The proposed SKP-ABE scheme counters a selectively chosen keyword attack game performed by an attacker if the DBDH assumption is valid. In the secure game model, attacker A can adaptively query the ciphertext for all search keywords. In that case, the plaintext of an index keyword is not exposed. In the security game model, it is assumed that probabilistic polynomial time attacker A and simulator B communicate with each other. Simulator B executes $Setup(1^\lambda)$ generates master key $(MK = <\alpha, \{t_i\}_{i\in[1,n]} >)$, and public parameter $(PP = <G, G_T, G_2\ e, g, \{T_i = g^{t_i}\}_{i\in[1,n]}, f = \quad g^k, EP = e(g,g)^\alpha, H_1, H_2 >)$, and sends PP to attacker A. A sends the attributes set $A = \{Att_1, Att_2, Att_3, \cdots, Att_n\}$; it wants to challenge to B. Attacker A requests the ciphertext index for the search keyword w' from B, and B outputs the ciphertext index (Index(w')). Then, it sends the output value to A. Attacker A selects two keywords w_0 and w_1 and sends them to B. B fairly selects a random bit of b ∈ {0,1}, has the attribute set $A = \{Att_1, Att_2, Att_3, \cdots, Att_n\}$ and w from the attacker, and outputs the corresponding value $I_{(w_b)} = <\tilde{C}_1, \tilde{C}_2, C_3 > \longleftarrow (\tilde{C}_1 = e(f, g^{w_b s'}), \tilde{C}_2 = g^{ss'}, C_3 = \prod_{i=1}^{n} H_1(Att_i)^s)$. Attacker A continuously requests a partial secret key query from B as in 2) and generates a final secret key. $(PSK = <g^\alpha, \{D_{i,1}\}_{i\in[1,n]}, H_{Att_i}> \longrightarrow FSK = <AS, D_i, D_i', \{D_{i,1}\}_{i\in[1,n]}, H_{Att_i} >)$. Then, by selecting the query keywords $w_{b'}$, a valid search token $T_{w_{b'}} = e(\prod_{i=1}^{n} H_{Att_i}, g^{w_{b'}})$ is continuously generated. Attacker A extracts b from $I_{(w_b)}$ with $T_{w_{b'}}$. However, it is difficult for an attacker to guess $b = b'$. Thus, the system is secure against selective chosen keyword attacks because the attacker finds it very difficult to win the game within probabilistic polynomial time. That is, it is difficult to guess the keyword plaintext information with the ciphertext index value created by Simulator B

(v) Protection against adaptively chosen plaintext attacks using a secure game model: The proposed SKP-ABE scheme counters an adaptively chosen plaintext attack game performed by an attacker if the DBDH assumption is valid. In the secure game model, attacker A can adaptively query the ciphertext for the selected plaintext and communicate with simulator B with each other. Simulator B executes $Setup(1^\lambda)$ generates MK and PP, the same as the chosen keyword attacks security game model. Attacker A requests a partial secret key query corresponding to the access structure $\{AS_1, \ldots, AS_n\}$ from B. At this time, the limitation is that the attribute set $A = \{Att_1, Att_2, Att_3, \cdots, Att_n\}$ must not satisfy the access structure $\{AS_1, \cdots, AS_n\}$. Attacker A receives the partial secret key $(PSK = <g^\alpha, \{D_{i,1}\}_{i\in[1,n]}, H_{Att_i}$

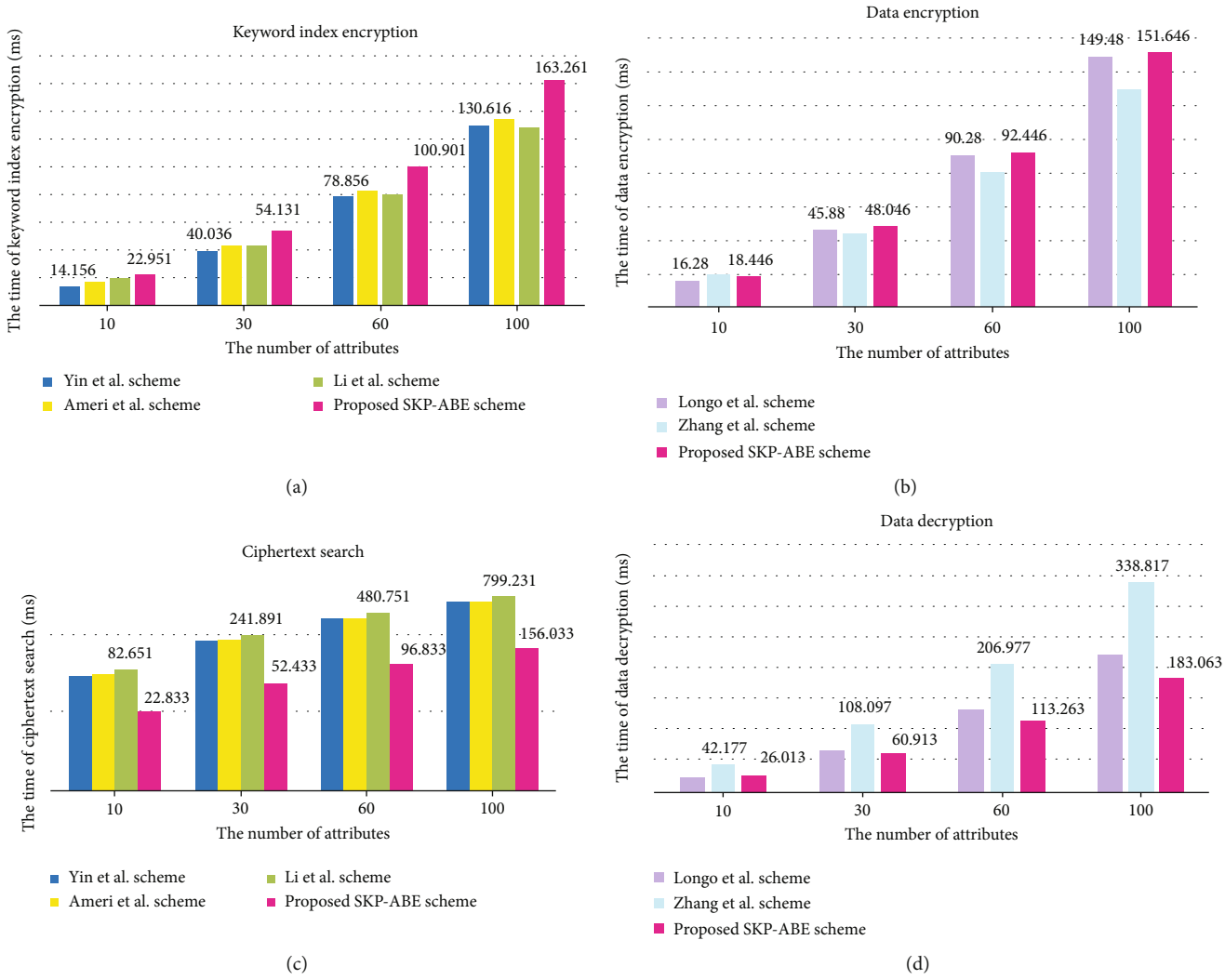


Figure 8: Comparison of the calculation amount of the existing KP-ABE scheme and this proposed scheme.

$>$) from B and generates a final secret key ($FSK = <AS, D_i, D_i', \{D_{i,1}\}_{i\in[1,n]}, H_{Att_i}>$). Attacker A selects two messages M_0 and M_1 and sends them to B. B fairly selects a random bit of $b \in \{0, 1\}$ and outputs the corresponding ciphertext $CT(M_b)$ with the attribute set $A = \{Att_1, Att_2, Att_3, \cdots, Att_n\}$ and w. And it sends the ciphertext $CT(M_b)$ to the attacker. Attacker A continuously requests the partial secret key query corresponding to the access structure $\{AS_1, \cdots, AS_n\}$ from B. Attacker A extracts b' from $CT(M_b)$. However, it is difficult for an attacker to guess $b = b'$. In other words, the system is selectively secure against adaptively chosen plaintext attacks, because the attacker finds it very difficult to win the game within the probabilistic polynomial time. It is difficult to guess the plaintext information through the ciphertext created by Simulator B

5.2. Efficiency. The computational amount measurements shown in Figure 8 were performed using a Windows system equipped with a 3.50GHz Intel Core i5-4690 processor and 8GB of RAM. Pairing calculations used the pairing-based cryptographic library available at [35]. ECC implementation used the Koblitz elliptic curve $y^2 = x^3 + ax + b(\text{mod } p)$ with $a = 1$ and $b = 1$ and the $163 - bit$ random prime defined as $F_{2^{163}}$. The proposed scheme includes a process of aggregating attributes in the encryption phase. Therefore, it can be seen from Figure 8 that the amount of computation required for keyword index encryption (a) and data encryption (b) is larger than that of the existing KP-ABE scheme. However, ciphertext search performance (c) and ciphertext decryption performance (d) are more efficient than the existing KP-ABE scheme. Therefore, the proposed SKP-ABE scheme efficiently provides ciphertext search and the user's ciphertext decryption performance. In order to compare the amount of computations in the same environment, one AA was assumed for the scheme of Longo et al. [13], and the scheme of Zhang et al. [14], when the calculation were performed.

(i) Efficient ciphertext search: When a user requests a ciphertext stored on the cloud server using keywords, search is generally inefficient because the

server decrypts all ciphertexts to find required data. Accordingly, we implement searchable encryption, which allows users to search for a requested ciphertext without having to decrypt the ciphertext. However, such schemes still suffer from several problems, as discussed above. Therefore, in our proposed SKP-ABE scheme, to address inefficient searching, the parameters of index I_w, that is, the attribute values corresponding to keywords in $C_3 = \prod_{i=1}^{n} H_1(Att_i)^k$, are aggregated and expressed a single value. The attribute values included in the token $T_{w'} = e(\prod_{i=1}^{n} H_1(Att_i)^k, g^{w'})$ are also aggregated and expressed as one value. Thus, if the attributes are {{Director}, {Company A}}, this can be expressed as {{Director}, {Company A}} = $C_3 = \prod_{i=1}^{n} H_1(Att_i)^k$. The ciphertext search seeks matches to C_3 regardless of the number of attributes. This is faster than the existing analyzed KP-ABE schemes because the number of searches is reduced as the number of attributes is irrelevant. Because the values of attributes are pre-aggregated, the user rapidly finds the required ciphertext

(ii) Constant-size ciphertext: In existing KP-ABE schemes, the size of the ciphertext increases in proportion to the number of attributes specified when generating a ciphertext. For example, in Yin et al.'s scheme, it can be seen through I(w) = $(A, I' = e(g_1, g_i)^{sH(w)}, I'' = g^s, \forall \alpha \in A : I_a = T(a)^s)$ that the size of the ciphertext increases according to the number of attributes in $\forall \alpha \in A : I_a = T(a)^s$. The size of the ciphertext varies depending on the attribute value I_a of 1 or a. In this proposed scheme, to provide a ciphertext of a constant size, the attribute values Att_i included in the ciphertext are aggregated and expressed as one value of $C_3 = \prod_{i=1}^{n} H_1(Att_i)^k$. Regardless of whether the number of attributes Att_i is 1 or i, all are all expressed as C_3. Therefore, it is possible to solve the problem that the number of existing attributes affects the size of the ciphertext. This only affects the ciphertext size, and since the attribute-based aggregation operation is performed in the data encryption phase, the disadvantage is that the amount of data encryption is large compared to the existing KP-ABE scheme

(iii) Efficiency of ciphertext decryption computations: In Table 3, several of the existing schemes (Yin et al., Ameri et al., and Li et al.) do not perform a decryption operation. Therefore, our proposed SKP-ABE scheme is compared with the scheme of Longo et al. and the scheme of Zhang et al., for decryption performance. As shown in Figure 8(d), the cost of decryption by the users is decreased compared to existing schemes (Longo et al., and Zhang et al.). Also, since the two schemes have the disadvantage that the decryption performance increases according to the number of AA, the efficiency of the proposed SKP-ABE scheme is better in terms of the user's decryption cost.

6. Conclusions

In this paper, we proposed an SKP-ABE system for secure and efficient data sharing in cloud environments. The proposed SKP-ABE scheme guarantees data confidentiality and integrity. Those who lack access rights are blocked. Specifically, the attribute value included in the token and the attribute value of the ciphertext are aggregated, and the ciphertext is searched using the aggregated value. As a result, since the number of ciphertext searches is not affected by the number of attributes, ciphertext searches can be performed quickly. Compared with the existing searchable KP-ABE schemes (Yin et al., Ameri et al., and Li et al.), the computation is efficient in terms of the number of ciphertext searches. In addition, when the data owner generates the ciphertext, the size of the ciphertext can be constant output without being proportional to the number of attributes by aggregating the values of the attributes included in the ciphertext. Finally, to solve the key escrow problem in AA, the user receives the PSK from the AA and generates the FSK in this proposed scheme. As a result, since the AA does not know information about the users' FSK, a key escrow problem cannot occur. Therefore, even if you try to decrypt the ciphertext stored in the cloud with only the users' PSK, data cannot be obtained. Compared to the existing KP-ABE scheme (Longo et al. and Zhang et al.) using multiple AA, the proposed scheme has better decryption performance efficiency.

The proposed SKP-ABE scheme is applied to N:1 cloud environment where a large number of data owners and a small number of data users share data. The scheme can be applied in various IoT-cloud environments, such as data sharing between nurses, doctors, and patients in a medical environment data sharing collected by drones in a UTM environment [36–38]. The shared data is secured because only authenticated users have access.

In the future, for the expansion of the proposed SKP-ABE scheme, additional research that can provide the requirements (security and efficiency) considered by KP-ABE is needed. Additionally, a signature and verification phase is required to decrypt the data user and verify that the owner uploads the data obtained.

Data Availability

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

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contributions

Yong-Woon Hwang and Su-Hyun Kim contributed equally to this work.

Acknowledgments

This research was supported by the Republic of Korea's MSIT (Ministry of Science and ICT), under the High-Potential Individuals Global Training Program (2021-0-01516) supervised by the IITP (Institute of Information and Communications Technology Planning & Evaluation), and this work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A2B5B01002490) and the Soonchunhyang University Research Fund.

References

[1] R. Kumar and R. Goyal, "On cloud security requirements, threats, vulnerabilities and countermeasures: a survey," *Computer Science Review*, vol. 33, pp. 1–48, 2019.

[2] A. Singh and K. Chatterjee, "Cloud security issues and challenges: a survey," *Journal of Network and Computer Applications*, vol. 79, pp. 88–115, 2017.

[3] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart Internet of Things systems: a consideration from a privacy perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[4] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 1–590, 2016.

[5] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP'07)*, pp. 321–334, Berkeley, CA, USA, 2007.

[6] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98, Alexandria, Virginia, USA, 2006.

[7] H. Yin, Y. Xiong, J. Zhang, L. Ou, S. Liao, and Z. Qin, "A key-policy searchable attribute-based encryption scheme for efficient keyword search and fine-grained access control over encrypted data," *Electronics*, vol. 8, no. 3, p. 265, 2019.

[8] M. H. Ameri, M. Delavar, J. Mohajeri, and M. Salmasizadeh, "A key-policy attribute-based temporary keyword search scheme for secure cloud storage," *IEEE Transactions on Cloud Computing*, vol. 8, no. 3, pp. 660–671, 2020.

[9] J. Li, M. Wang, Y. Lu, Y. Zhang, and H. Wang, "ABKS-SKGA: attribute-based keyword search secure against keyword guessing attack," *Computer Standards & Interfaces*, vol. 74, no. 103471, pp. 103471–103477, 2021.

[10] R. Meng, Y. Zhou, J. Ning, K. Liang, J. Han, and W. Susilo, "An efficient key-policy attribute-based searchable encryption in prime-order groups," in *International Conference on Provable Security*, pp. 39–56, Xi'an, China, 2017.

[11] C. J. Wang and J. F. Luo, "A key-policy attribute-based encryption scheme with constant size ciphertext," in *2012 Eighth International Conference on Computational Intelligence and Security*, pp. 447–451, Guangzhou, China, 2012.

[12] J. Lai, R. H. Deng, Y. Li, and J. Weng, "Fully secure key-policy attribute based encryption with constant-size ciphertexts and fast decryption," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*, pp. 239–248, Kyoto, Japan, 2014.

[13] R. Longo, C. Marcolla, and M. Sala, "Collaborative multi-authority KPABE for shorter keys and parameters," *IACR Cryptology. ePrint Archive*, vol. 262, pp. 1–23, 2016.

[14] L. Zhang, P. Liang, and Y. Mu, "Improving privacy-preserving and security for decentralized key-policy attributed-based encryption," *IEEE Access*, vol. 6, pp. 12736–12745, 2018.

[15] Y. Song, H. Wang, X. Wei, and L. Wu, "Efficient attribute-based encryption with privacy-preserving key generation and its application in industrial cloud," *Security and Communication Networks*, vol. 1155, Article ID 3249726, 2019.

[16] H. Wang, X. Dong, and Z. Cao, "Multi-value-independent cipher text policy attribute-based encryption with fast keyword search," *IEEE Transactions on Services Computing*, vol. 13, no. 6, pp. 1142–1151, 2020.

[17] H. Wang, X. Dong, Z. Cao, and D. Li, "Secure and efficient attribute-based encryption with keyword search," *The Computer Journal*, vol. 61, no. 8, pp. 1133–1142, 2018.

[18] Y. W. Hwang and I. Y. Lee, "A study on CP-ABE-based medical data sharing system with key abuse prevention and verifiable outsourcing in the IoMT environment," *Sensors*, vol. 20, no. 17, p. 4934, 2020.

[19] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[20] C. Hu, J. Yu, X. Cheng, Z. Tian, and L. Sun, "CP-ABSC: An attribute based signcryption scheme to secure multicast communications in smart grids," *Mathematical Foundations of Computer Science*, vol. 1, no. 1, 2018.

[21] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext policy attribute-based encryption scheme with constant ciphertext length," in *International Conference on Information Security Practice and Experience (ISPEC)*, pp. 13–23, Xi'an, China, 2009.

[22] J. Kim, W. Susilo, F. Guo, M. H. Au, and S. Nepal, "An efficient KP-ABE with short ciphertexts in prime order groups under standard assumption," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 823–834, Abu Dhabi, United Arab Emirates, 2017.

[23] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceeding 2000 IEEE symposium on security and privacy*, pp. 44–55, Berkeley, CA, USA, 2000.

[24] Y. W. Hwang and I. Y. Lee, "A study on data sharing system using ACPABE-SE in a cloud environment," *International Journal of Web and Grid Services*, vol. 17, no. 3, pp. 201–220, 2021.

[25] G. K. Verma, B. B. Singh, N. Kumar, and V. Chamola, "CB-CAS: certificate-based efficient signature scheme with compact aggregation for industrial internet of things environment," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2563–2572, 2019.

[26] K. A. Shim, "A new certificateless signature scheme provably secure in the standard model," *IEEE Systems Journal*, vol. 13, no. 2, pp. 1421–1430, 2018.

[27] X. Zhang, C. Jin, Z. Wen, Q. Shen, Y. Fang, and Z. Wu, "Attribute-based encryption without key escrow," in *International Conference on Cloud Computing and Security*, pp. 74–87, Nanjing, China, 2015.

[28] M. Chase, "Multi-authority attribute based encryption," in *Theory of cryptography conference*, pp. 515–534, Berlin, Germany, 2007.

[29] M. P. Hoyle and C. J. Mitchell, "On solutions to the key escrow problem," in *State of the Art in Applied Cryptography*, pp. 277–306, Leuven, Belgium, 1998.

[30] K. Zhang, J. Gong, S. Tang et al., "Practical and efficient attribute-based encryption with constant-size ciphertexts in outsourced verifiable computation," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pp. 269–279, Xi'an, China, 2016.

[31] S. Belguith, N. Kaaniche, and G. Russello, "PU-ABE: Lightweight attribute-based encryption supporting access policy update for cloud assisted IoT," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pp. 924–927, San Francisco, CA, USA, 2018.

[32] J. Li, S. Cheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Approximate holistic aggregation in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 13, no. 2, pp. 1–24, 2017.

[33] M. Chase and S. S. M. Chow, "Improving privacy and security in multiauthority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security*, pp. 121–130, Chicago Illinois, USA, 2009.

[34] Q. Li, H. Xiong, F. Zhang, and S. Zeng, "An expressive decentralizing kp-abe scheme with constant-size ciphertext," *International Journal of Network Security*, vol. 15, no. 3, pp. 161–170, 2013.

[35] B. Lynn, "The pairing-based cryptography (PBC) library," 2010, http://crypto.stanford.edu/pbc.

[36] L. Touati and Y. Challal, "Collaborative kp-abe for cloud-based internet of things applications," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–7, Kuala Lumpur, Malaysia, 2016.

[37] S. Y. Tan, K. W. Yeow, and S. O. Hwang, "Enhancement of a lightweight attribute-based encryption scheme for the Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6384–6395, 2019.

[38] B. Girgenti, P. Perazzo, C. Vallati, F. Righetti, G. Dini, and G. Anastasi, "On the feasibility of attribute-based encryption on constrained IoT devices for smart systems," in *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 225–232, Washington, DC, USA, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8022033, 11 pages
https://doi.org/10.1155/2022/8022033

Research Article

HomeGuardian: Detecting Anomaly Events in Smart Home Systems

Xuan Dai,[1] **Jian Mao**,[1,2] **Jiawei Li**,[1,2] **Qixiao Lin**,[1,2] **and Jianwei Liu**[1]

[1]*School of Cyber Science and Technology, Beihang University, 37 Xueyuan Road, Haidian District, Beijing, China 100191*
[2]*Beihang Hangzhou Innovation Institute Yuhang, Xixi Octagon City, Yuhang District, Hangzhou, China 310023*

Correspondence should be addressed to Jiawei Li; daweix@buaa.edu.cn

Received 29 April 2022; Accepted 23 May 2022; Published 13 June 2022

Academic Editor: Yan Huang

As a typical application of Internet of Things (IoT), home automation systems, namely, smart homes, provide a more convenient and intelligent life experience through event recognition, automation control, and remote device access. However, smart home systems have also given rise to new complications for security issues. As an event-driven IoT system, smart home environments are vulnerable to security attacks, and vulnerable devices are far-spread due to the quick development cycles. Attack vectors to smart homes inevitably manifest in abnormal event contexts. In this paper, we propose HomeGuardian, a context-based approach to identify abnormal events in smart homes. In our approach, we extract temporal context and environmental context from system logs, aggregate (embed) these hybrid contexts, and construct a learning-based classifier to identify the abnormal events. We develop a testbed to implement and evaluate our approach.

1. Introduction

Smart home, as a ubiquitous computing IoT application in a home environment [1–3], provides remote control and automation services for home users. Quick development cycles of smart devices lead to an increasing expansion of the smart home market scale [4–6].

However, smart home industry develops rapidly without neither a unified security standard nor a unified supervision mechanism, and the inconsistency leads to many security problems. Once the devices are accessed illegally by attackers, users' privacy suffers severe leakage [7, 8]. Moreover, if the smart home hub is accessed by attackers, he/she may obtain control privileges (e.g., door locks) and implement data interception or workflow interference. SmartThings exposes over 20 vulnerabilities in its hub [9]. In addition to hardware vulnerabilities, malicious software in smart homes also results in security and privacy issues. Malicious smart home applications can steal private information by obtaining nonnecessary permissions [10]. In addition to controlling devices directly, physical interactions and automation rules also introduce security risks in smart homes [11, 12]. These security issues will inevitably lead, directly or indirectly, to abnormal device state changes (i.e., events).

Since abnormal events are the most intuitive manifestation of abnormalities in the visible aspect, researchers focus on analyzing event sequences (i.e., device state changes) in the smart home. Hidden Markov Model (HMM) is widely used to analyze sequences for abnormal event detection in smart homes [13]. Event correlation analysis is also applied in smart home scenarios [14, 15]. However, these methods only take into account sequence order without considering specific timing information of smart home events. Besides, in IoT systems, smart devices interact with each other through automation rules and physical effects. Therefore, the states of surrounding devices also should be considered to validate the device behaviors.

To solve the above problems, we propose an anomaly detection system based on event context. In our approach, we take two types of context into account, *temporal* context and *environmental* context.

The *temporal* context of a candidate event is based on the event's time intervals and the history device states. We

further predict the successive events as the *temporal* context to feed HomeGuardian. The *environmental* context for an event is the states of the devices related to the event, which indicates the physical context of the candidate event. Specifically, HomeGuardian analyzes correlations among devices and selects the states of devices that are highly correlated to the candidate event as its *environmental* context features. Given the two contexts, HomeGuardian then implements a difference-based anomaly detection, by inspecting whether the input event is expected, considering the two contexts.

To sum up, we make the following contributions:

(i) We propose HomeGuardian, a context-based approach to identify abnormal events in a smart home system. In our approach, we extract temporal context and environmental context from system log, aggregate (embed) these hybrid contexts, and construct a learning-based classifier to identify the abnormal events

(ii) We develop a self-configured testbed based on the Home Assistant platform to implement and evaluate our approach. According to real-world smart home scenarios, we connect virtual devices with the real hardware environment and configure automation rules to simulate/generate smart home event data

(iii) We evaluate the effectiveness of HomeGuardian using the system log captured from our self-developed testbed. The experiment results illustrate that the F1-scores of HomeGuardian to detect abnormal events are above 0.90 for all device types

2. Background and Problem Statement

2.1. Security Problems of Smart Home. Smart device state changes triggered by automation rules or remote control often cause security risks. For instance, user-setup rules can be triggered accidentally. Figure 1 illustrates how an attacker opens a window and breaks in when nobody is home via triggering rule IF *Temperature* > 25 THEN *open* the *window*. Moreover, a smart home may catch fire if heating devices (e.g., ovens and electric heaters) are turned on by abnormally triggered automation rules.

Vulnerable smart devices may be controlled by an attacker remotely to launch abnormal events or distort device states [9]. The related attacks include network penetration, firmware backdoor exploitation [16], replay attacks [17, 18], and man-in-the-middle attacks [19, 20]:

(1) http.//[Router_IP]/…&SystemCmd = [Malicious_ Code]&…

(2) Heater. off ⟶ on

(3) Temperature sensor. 23 ⟶ 27

(4) If (*temperature* > 25) then window. $off \longrightarrow on$

In addition to devices, smart applications (apps for short) also contribute to system vulnerabilities. Without the knowledge of users' environment settings and behavioral patterns [21], it is challenging for smart apps to precisely define a condition, such as "at home," because the hardware/software settings and user behavioral patterns (e.g., wake-up time, bedtime, and time to take a shower) vary in homes.

2.2. Problem Analysis. To detect anomalies in smart home systems, most existing approaches focus on program analysis for platforms and apps while overlooking device interactions, which are leveraged in real-world attacks like the scenario in Figure 1. Hence, additional contexts such as device states are required for robust and noninvasive anomaly detection.

As shown in Table 1, the correlations of state changes (i.e., events) include objective environmental changes, user behavior patterns, influence between devices, and automation rules for system configuration in a smart home system. Automation rules and smart apps manipulate devices to meet user demands. Besides, a device can be affected by a physical channel between another device. For example, an air conditioner has an impact on an adjacent thermometer. User behavior and environmental changes (e.g., day-night cycle or season alternation) can also cause periodic changes in states of thermometers, hygrometers, and other devices.

However, it is challenging to extract environmental correlation in a smart home system. Static-analysis-based methods cannot capture physical environmental interactions among devices. IoTMon [11] analyzes the description of IoT apps to recognize common physical channels between IoT devices and discover potential correlations between devices and the environment. Nevertheless, it is not effective to detect real-world runtime physical environmental interaction influences.

Hence, robust anomaly detection should consider the time at which devices interact through physical channels. Moreover, some interactions between devices and the physical environment may occur either immediately (e.g., turn on a light) or slowly and continuously (e.g., boil water via a heater), temporal context of the smart home matters when detecting anomalies. To sum up, we should implement a systematical analysis when detecting smart home anomalies, taking both environmental context and temporal context into consideration.

3. System Design

3.1. Overview. When analyzing the correlation of smart home devices, we consider the influence between automation rules and devices. We first extract the correlated device states as environmental context. We pinpoint the environmental states when target events are triggered and then build feature vectors. For device behavior regularity caused by user behavior and environmental changes, we extract the state changes of target devices from the logs, model the behavior, and refine the temporal context to forecast the next event. Finally, we construct a classifier and implement the abnormal detection systems in smart homes.

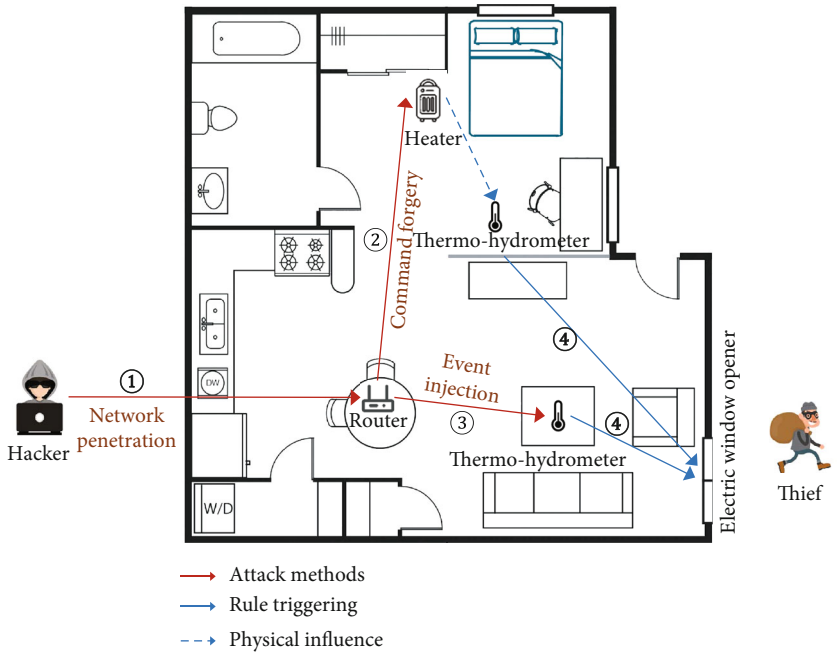


Figure 1: An example scenario where an attacker breaks into smart home to open the window.

Table 1: IoT device influences.

Influence	L	Source	Processing by
Automation rules	A	Automation rules and apps	System analysis
Influences among devices	P	Physical influences	
User behavior	A	User commands	Behavior modeling
	N	Network traffic	
	P	User activities	
Environmental changes	P	Periodic changes	

Note: L: layer; A: app layer; N: network layer; P: physical layer.

The framework of the anomaly detection system HomeGuardian is shown in Figure 2. The purpose of HomeGuardian is to screen out abnormal events from smart home platform logs. It is composed of three modules, namely, *testbed platform*, *context extraction*, and *anomaly detection*. Specifically, the testbed platform is a smart home experimental platform with multiple functions, such as device control and behavior simulation. The context extraction module extracts the environmental context among smart devices by analyzing the device configurations and rule configurations of the smart home platform. Besides, it also uses machine learning algorithms for behavior modeling to predict the following state of the target device. Environmental context and temporal context are the input of the anomaly detection module. The anomaly detection module is composed of a neural network, which filters out abnormal events.

3.2. Temporal Context. The smart home events recorded in platform logs can be represented as (timestamp, device, state), namely, the date and time when the device state changes occur, the device name, and its state value after the event occurs, respectively. Note that apart from successive values (e.g., temperature and humidity), states can also be presented in binary values (e.g., ON/OFF for a switch and OPEN/CLOSE for a door). A log sample reads as $(2021-05-1108:58:31, light.L001, on)$

To obtain the temporal contextual feature of the smart home events, the log entries corresponding to the target device, i.e., the state changes of the target device, are first filtered out from the log. The time interval of event occurrence is calculated based on the event timestamps. The change of the time interval length reflects the frequency of the target device events with the event change pattern. We determine the analysis method by collecting data from the smart home testbed platform. To keep the consistency of time increments, we denote the first presence of a certain event with the timestamp timestamp_0 as an initial event e_0 and then calculate the time interval between e_0 and the subsequent events $e_i|_{i=1,2,\ldots}$ as $t_i = \text{timestamp}_i - \text{timestamp}_0$, where timestamp_i is the timestamp for event e_i. In this way, the temporal context features are monotonically increasing in chronological order.

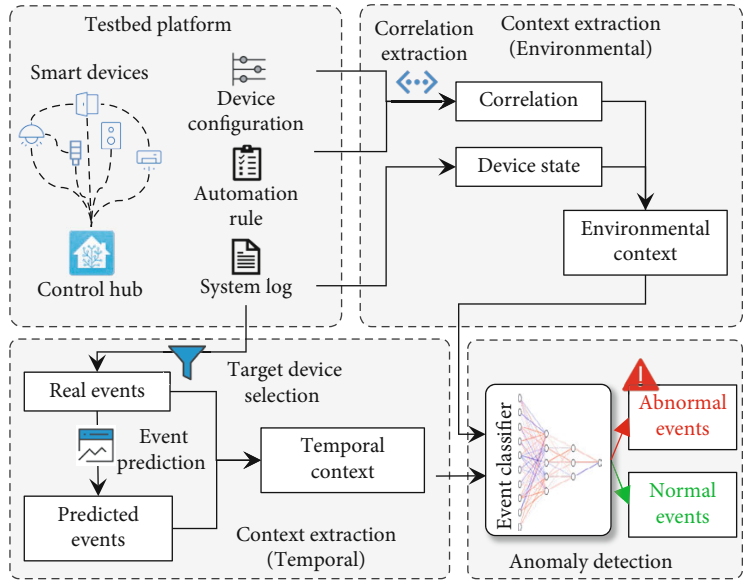


Figure 2: Framework of HomeGuardian.

After that, a learning model is constructed based on the features extracted from the log. It predicts the moment and state value of the next state change of the target device. The inputs of the model include the prior N states of the target device $\mathbf{s} = (s_1 ; s_2 ; \cdots ; s_N)$, and the prior N occurrence times of the target device $\mathbf{t} = (t_1 ; t_2 ; \cdots ; t_N)$. Among them, the parameter $\mathbf{t}$ is the record starting point at the specified time, which can be updated in a period to prevent the data from being too large. The output of the model is denoted as (t^P, s^P). t^P is the time interval of the next event, and s^P is the state of the next event. We apply a regression algorithm to extract the relationships among input variables. The outputs are presented as $t^P = \alpha_1^T \mathbf{t} + \delta_1 + \varepsilon_1$ and $s^P = \alpha_2^T \mathbf{s} + \delta_2 + \varepsilon_2$, where α_1 and α_2 are the weight coefficient vectors of each feature vector, i.e., how closely each data quantity is associated with the device state value. δ_1 and $\delta_2$are constant terms of intercepts. ε_1 and ε_2 are errors obeying a normal distribution with a mean of 0. t^P and s^P are predictions of the time of the next event and the state of the device for the same device. The temporal context $\mathbf{f}_t$ includes the time prediction t^P and state prediction s^P for subsequent events and the actual time t^R and actual state t^R of the event occurrence. It can be presented as $\mathbf{f}_t = (t^P ; s^P ; t^R ; s^R)$.

3.3. Environmental Context. The environmental context of system events refers to the device states correlated to the target device. The correlations come from user-defined automation rules, physical channels, and spatial relationships between devices. An important characteristic of smart homes is that smart devices may cause impacts on the physical environment. Such physical influence brings the correlation of state changes between devices [11]. Further, there are interactions between user-defined automation rules and IoT applications. Physical channels, such as temperature, humidity, and brightness, enable devices with certain attributes to interact with each other. For example, there is an automation rule that controls a radiator to turn on or off according to room temperature. In this case, the temperature channel connects the heater and temperature sensor with correlation.

Based on the aforementioned analysis, relevant device selection is determined according to automation rules. The spatial distribution of the devices and the physical channels shared among the devices impacts their correlation as well. The correlation degree $R_{D,C}$ between a device D and a physical channel C is obtained by Natural Language Processing (NLP), which refers to device correlations. Then, we obtain the semantic similarities $R_{D,C}$ of each device name words and physical channels in smart homes. We use a threshold Θ_R and consider the devices whose $R_{D,C} > \Theta_R$ as containing the *physical property* C.

The devices with the same physical channel and spatially-near locations are considered correlated. Meanwhile, the devices affected by the automation rule are also considered correlated. Then, we get the correlation matrix $\mathbf{G}$.

During the training phase of our model, since the data collected from smart home logs only includes the state change of devices, it is necessary to maintain a cache matrix variable $\mathbf{M}_{L\times N}$ to record the current state of all devices in the environment. Each row of the matrix represents the previous N state values of the target device, and each column represents the state of all L devices in the current time $\mathbf{s}_D = (s^{D1} ; s^{D2} ; \cdots ; s^{DL})$, i.e., the environment state. In this step, we select k other devices with the highest correlation of target devices for the following calculation. First, we obtain the data related to the target event for anomaly detection, including states of all L devices in the environment, and the corresponding correlation vector which is denoted as $\mathbf{g} = \mathbf{G}_{*,j}$. $\mathbf{g}$ is the correlation vector between the target device and other devices, and the values of each item are 0 or 1. (i.e., a column in correlation matrix $\mathbf{G}$). The devices with

correlation are marked as 1, and others are marked as 0. The Hadamard product of $\mathbf{g}$ and $\mathbf{s}_D$ is the device state value vector correlated with the target equipment. Furthermore, we remove the zero values and reduce the dimension of the state value vector. After the above steps, we obtain the current state changes correlated to the target devices, i.e., the environmental context features, which are denoted as $\mathbf{f}_e = (s^{D1}; s^{D2}; s^{D3}; \cdots; s^{Dk})$.

Algorithm 1 summarizes the above environmental context feature extraction process.

3.4. Event Classifier. Given $\mathbf{f}_t$ and $\mathbf{f}_e$, we utilize Neural Networks (NN) to classify normal events and abnormal events. According to the characteristics and experience of the target problem, if the NN has two hidden layers and an appropriate activation function, it can fit any decision boundary or smooth mapping with any accuracy [22]. Since the context feature may differ for different types of candidate events, for example, some events may pay more attention to the historical trend, some pay more attention to the surrounding environment, and some need to be comprehensively consider these two factors. In order to consider the impact of time and environmental context for event classification at the same time, we concatenate the two feature vectors as NN input, and we learn the weight relationship of each feature to the judgment result by training our NN.

The input of our NN is vector $\mathbf{x} \in R^{k+4}$ concatenated from temporal context $\mathbf{f}_t \in R^4$ and environmental context $\mathbf{f}_e \in R^k$, i.e., $\mathbf{x} = \{\mathbf{f}_t, \mathbf{f}_e\}$. The output is the judgment result $\hat{y}$. We use ReLU on the hidden layers and Sigmoid on the output layer to map the result to $[0, 1]$. When training, the normal events are marked as 0, abnormal events are marked as 1 , and we set a threshold on the result to give judgment. The input layer has n nodes, there are $n/2$ nodes in the first hidden layer, and 3 nodes in the second hidden layer. Temporal context $\mathbf{f}_t$ includes predicted time t_i^P, predicted value s_i^P, real time t_i^R, and real value s_i^R, which describes deviations between predicted and true values. The environmental context $\mathbf{f}_e$ includes the states of the selected k devices, which consist of related device states $s_i^{D1}, s_i^{D2} \cdots s_i^{Dk}$. The forward propagation process of our NN is described as $\mathbf{h}_1 = ReLU(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$, $\mathbf{h}_2 = ReLU(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2)$, and $\hat{y} = \text{Sigmoid}(\mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3)$, where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3$ refer to the weight matrices for each layer, and $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ refer to the layer bias vectors. We use binary crossentropy (BCE) [23] as our loss function, which is denoted as $L(y, \hat{y}) = -y \cdot \log \hat{y} + (1 - y) \cdot \log (1 - \hat{y})$, where $\hat{y}$ refers to the probability to predict samples as positive of our model, i.e., the probability of an event to be predicted as abnormal. y refers to the sample label, which is 1 when the sample is positive and 0 when negative. We adopt a binary classification method to determine whether there are abnormal events in the smart home system. Regular device state changes in a smart home are generally normal events. The training process requires not only the event log under a normal environment but also abnormal events. To achieve this, we obtain the abnormal data from our self-designed testbed by simulation. The features of normal and abnormal events are extracted based on the log generated by the smart home platform.

4. Implementation

4.1. Simulation-Based Data Collection. Our anomaly detection system is deployed on a heterogeneous system with different brands of devices connected to Home Assistant. Normal behavior is obtained directly from the system logs. To simulate abnormal behavior, we reproduce an injection attack for forgery sensor event. In particular, we intercept the token through a man-in-the-middle attack via a ZigBee gateway and forge POST requests from sensors to the Home Assistant server to overwrite the states of real devices. After injecting the forgery event, normal and abnormal events are indistinguishable in platform system logs. Thus, we capture the records of events replied by the Home Assistant to mark the events injected by our attack.

After processing the data collected from the virtual and real smart home platforms, we observe that the time interval between two events varies randomly. Thus, this feature is inappropriate as a criterion for effective detection. At the same time, if there are unexpected situations such as network disconnection and system downtime in the smart home system, the time interval of events will increase and drop sharply, which might seriously affect the accuracy of our behavior model. Therefore, in this work, the event time information is uniformly converted to Unix timestamp [24]. Through actual testing, we found that the frequency of event occurrence is different for different devices. Therefore, an appropriate start time could be selected according to the device types. Take motion sensors for instance, if the event frequency is high, the first state change moment of each week could be selected as the starting time point. Differently, temperature sensors have low-frequency events, and the first state change every three months could be selected as the starting point as seasonal effects need to be considered.

4.2. Environment Association Extraction. In this step, we use word2vec [25] to convert the extracted keywords into two vectors, namely, V_D (device name word vector) and V_C (physical channel semantics vector) and then calculate their cosine similarity, which is given by $R_{D,C} = \mathbf{v}_D \cdot \mathbf{v}_C / \|\mathbf{v}_D\| \cdot \|\mathbf{v}_C\| = \sum_{i=1}^{n} v_{Di} \times v_{Ci} / \sqrt{\sum_{i=1}^{n} v_{Di}^2} \times \sqrt{\sum_{i=1}^{n} v_{Ci}^2}$. We use the word vector model from Google News [26] as a pre-trained corpus. We extract keywords of device entities in Home Assistant (e.g., "light," "sensor," and "door") and use word2vec to embed them. The correlation between devices and physical channels is represented by the cosine similarity between device keywords and physical channel keywords (e.g., "motion," "illuminance," and "sound").

Based on the established rule set and semantic association information, we obtain the association table of IoT devices. As shown in Table 2, the relevance includes deterministic association rules in which the trigger condition involves the correlation between the monitoring device and the target device to carry out the instruction. These associations are determinate because as long as the trigger condition is met, the smart home system will instruct target devices to perform corresponding operations. It is important to note that associations through physical channels need to combine with the spatial location of devices.

Input: n_D, n_C, Θ_R, $\mathbf{s}_D$
Output: $\mathbf{f}_e$
/* compute similarity of word vector, build correlation matrix $\mathbf{G}$ */
for $i = 0 \longrightarrow len(n_C)$ **do**
  for $j = 0 \longrightarrow len(n_D)$ **do**
    $R_{D,C} = similarity(n_C, n_D)$
    if $R_{D,C} > \Theta_R$ **then**
      $\mathbf{G}_{i,j} \longleftarrow 1$
    else
      $\mathbf{G}_{i,j} \longleftarrow 0$
/* build feature vector */
$\mathbf{f}_D \longleftarrow \mathbf{G}_{*,j} \mathbf{es}_D$
/* remove 0 values*/
for *item* **in** $\mathbf{f}_D$ **do**
  if $item \neq 0$ **then**
    $\mathbf{f}_e.append(item)$
return $\mathbf{f}_e$

Algorithm 1: Environmental Context Feature Extraction.

Table 2: Correlation between smart home devices.

Causality	Code	a	b	c	d	e	f	g	h	i
Temperature sensor	a	—	—	√	√	—	—	—	—	—
Humidity sensor	b	—	—	√	—	√	—	—	—	—
Air conditioner	c	○	○	—	√	√	—	—	—	—
Smart socket	d	—	—	√	—	—	—	—	—	—
Heater	e	○	○	√	—	—	—	—	—	—
Humidifier	f	○	○	√	—	—	—	—	—	—
Wireless switch	g	—	—	—	—	—	√	—	—	—
Bedside lamp	h	—	—	—	—	—	—	—	○	—
Night light	i	—	—	—	—	—	—	—	○	—
Motion sensor	j	—	—	—	—	—	—	√	—	—
Door and window sensor	k	—	—	—	—	—	—	—	—	√

√: association rules triggered; ○: environment triggered.

During data processing, we record the latest N states of each device in smart homes (according to device type, N is selected from 2, 10 in our implementation). For each state, we also require the current states of other devices (i.e., the environment state). In the detection phase, HomeGuardian then implements real-time anomaly detection by capturing device states in the current environment from the Home Assistant platform through GET requests.

5. Evaluation

5.1. Experiment Setup. To collect testing event data, we develop a self-configured testbed based on the Home Assistant platform. Our testbed supports virtual device simulation, which is based on HH114 dataset from CASAS [27]. The testbed consists of real smart devices and simulated virtual devices. Event data related to experiments are logged and can be extracted on-demand.

We mainly focus on four types of devices below: motion sensors, temperature sensors, lights, and illumination sensors. The virtual devices can also be bound to devices in the laboratory, which can reflect the physical interactions (e.g., interactions between a virtual light L001 and a real illumination sensor LS001.) Figure 3 shows the layout of virtual devices in a room map in our experiments. Environmental factors influence devices located in the same color area. The devices marked with an asterisk are real devices, and devices with two asterisks are simulated virtual devices.

By sending events to the testbed platform iteratively, we simulate logs containing interactions produced by preset custom automation rules. Since events in the CASAS dataset are captured under a real scenario that can reflect user behaviors, simulated logs are largely consistent with the real logs. The logs can be directly obtained from the Home Assistant platform if connected devices exist. The virtual devices are used as supplementary. Users manually manipulate or use external scripts to control the real devices and record real events.

5.2. Effectiveness on Event Prediction. In this subsection, we first determine the count of historical events N selected

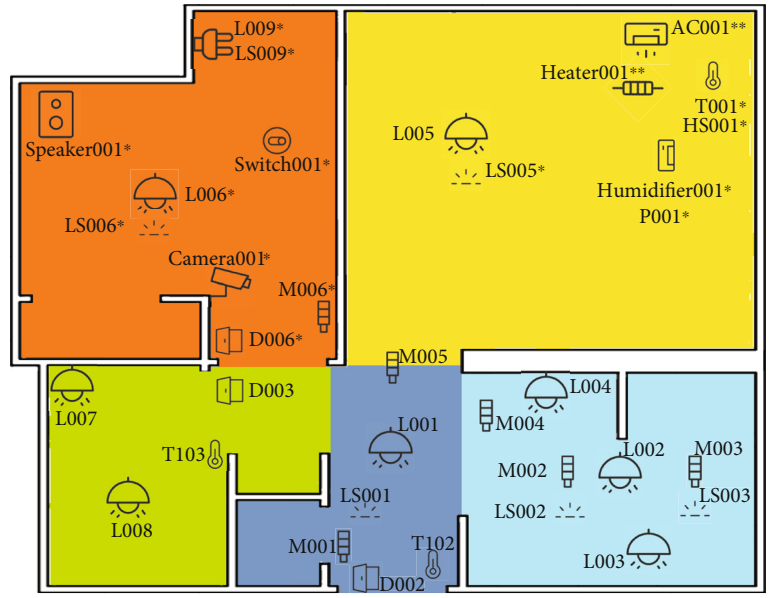


Figure 3: Device deployment layouts of testbed.

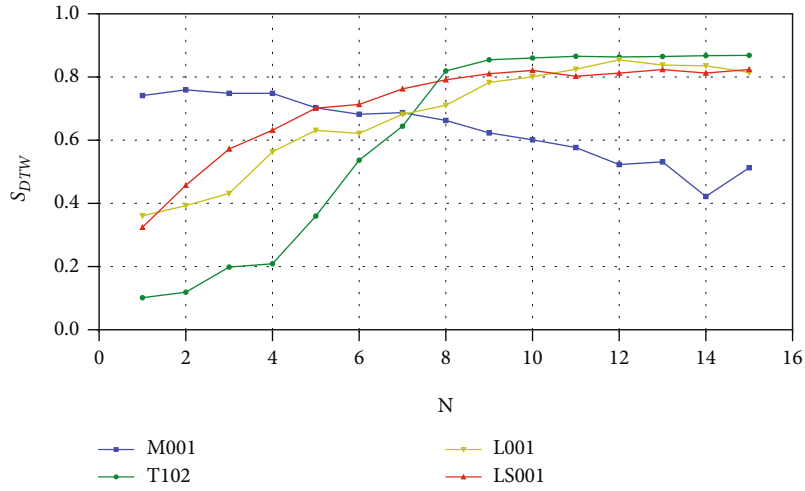


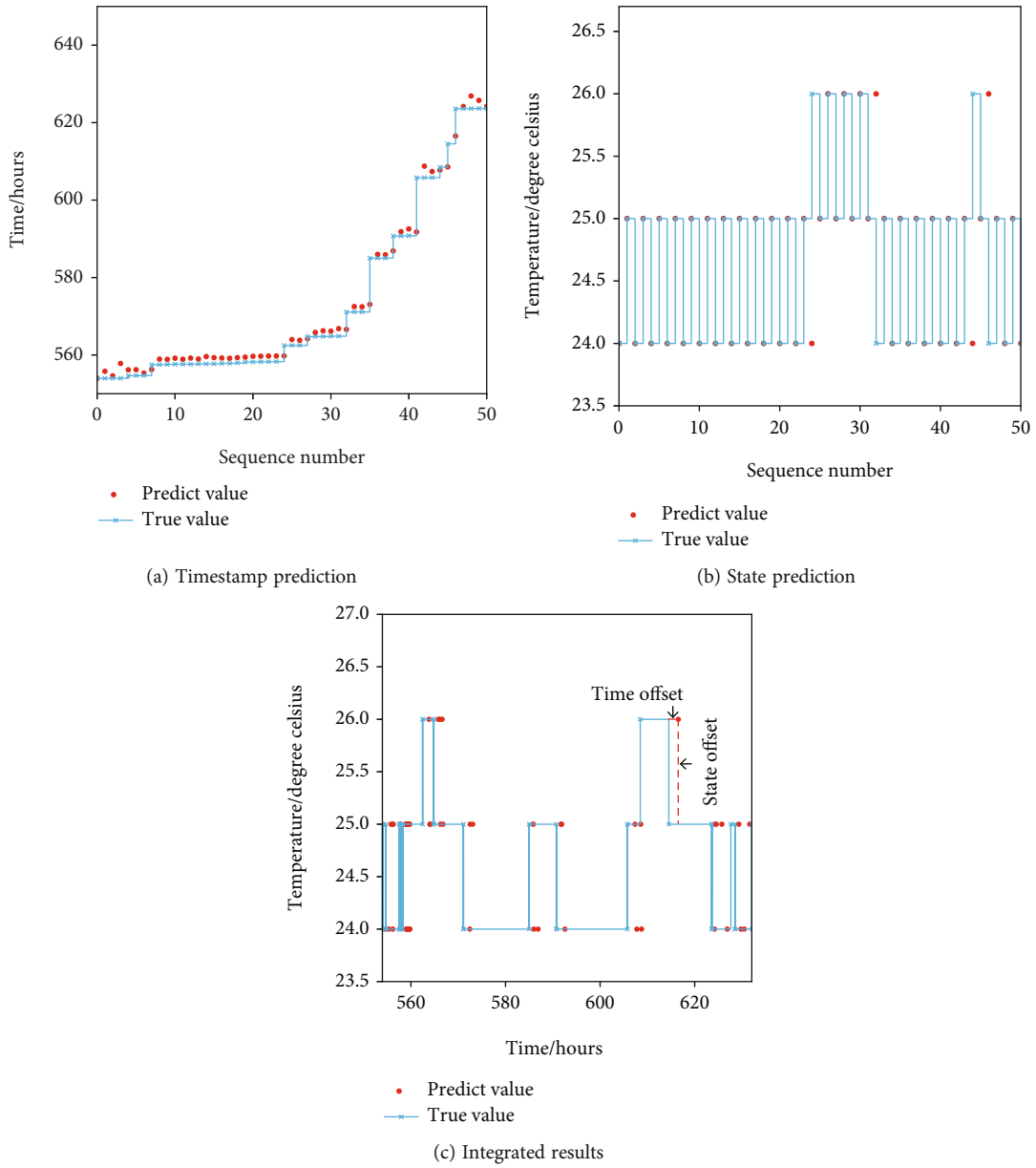
Figure 4: The influence of historical data length N on event prediction results.

when predicting the target one and evaluate the effectiveness of the event prediction mechanism. We select four types of typical devices for the following test.

5.2.1. Parameter Selection. We use dynamic time warping (DTW for short) as an algorithm measuring the distance between a couple of time series and calculate similarity distances L_{DTW} between a predicted sequence A' and a real sequence A. Thus, the similarity of these sequences is calculated as $S_{\mathrm{DTW}} = 1 - 2\mathrm{L}_{\mathrm{DTW}}/\mu_A + \mu_{A'}$. Where μ represents the mean state value of each sequence. The similarity is then used as an evaluation benchmark for prediction effectiveness. As shown in Figure 4, we can observe the impact of N selection on event prediction results by calculating S_{DTW} of event predictions for four different devices with incremental N values selected.

For motion sensors, the prediction effectiveness decreases as the history length N increases as shown by the blue discount, which means these device states are not strongly correlated with historical data. Therefore, we pay more attention to the environmental context when anomaly detection, such as changes under correlation between each motion sensor and its surrounding motion sensors in real scenarios. Consequently, we take $N = 2$ for devices of this kind. For temperature sensors, illumination sensors, and lights, the prediction effect will not achieve the best goal until N is about 9 to 10, and it only fluctuates slightly with N larger than 10. Accordingly, we take $N = 10$ for data processing.

5.2.2. Results. We divide the first 80% of the dataset in the specified time interval as the training set and the rest of the dataset as the testing set. Rest experiments mentioned in this paper all take the same method. Taking the temperature sensor T102 as an example, the prediction results of its state changes are shown in Figures 5(a) and 5(b). The cyan



(a) Timestamp prediction

(b) State prediction

(c) Integrated results

Figure 5: Prediction results of temperature sensor T102.

Table 3: Event prediction NRMSE.

Device	M001	M002	T102	T103	L001	L002	LS001	LS002
$NRMSE_t$	0.011	0.014	0.023	0.025	0.028	0.021	0.021	0.023
$NRMSE_s$	0	0	0.1394	0.1648	0	0	0.051	0.060

Note: $NRMSE_t$: NRMSE of timestamps; $NRMSE_s$: states prediction.

data represents the true value, the red data represents the predicted value, and the fitting effect is as expected.

Exceptionally, the output values of the model need to be adjusted to the specifics of different IoT devices. Since the accuracy of the device T102 thermometer is 1 degree Celsius in the current dataset, the output of the model can be rounded. Binary value devices, such as human motion sensors and light switches, output in the form of ON/OFF. The subsequent output value must be the inverse of the current state. Thus, the binary value device only needs to predict the moment of the next occurrence, without considering device states.

While using event timestamps as the x-coordinate and the state value of the device as the y-coordinate, the predicted events and the real events are put together for comparison. The moment offset and state offset of the

Table 4: Abnormality classifier effectiveness for each device.

Device	Temporal context only			Environmental context only				Single-output NN				Dual-output NN
	Precision	Recall	F1-score	Precision	Precision	Recall	F1-score	Precision	Precision	Recall	F1-score	Precision
M001	0.84	0.92	0.88	0.69	0.84	0.92	0.88	0.69	0.84	0.92	0.88	0.69
M002	0.85	0.88	0.86	0.48	0.85	0.88	0.86	0.48	0.85	0.88	0.86	0.48
T102	0.93	0.89	0.91	0.41	0.93	0.89	0.91	0.41	0.93	0.89	0.91	0.41
L002	0.52	0.63	0.57	0.99	0.52	0.63	0.57	0.99	0.52	0.63	0.57	0.99
LS001	0.57	0.39	0.46	0.97	0.57	0.39	0.46	0.97	0.57	0.39	0.46	0.97
LS002	0.41	0.28	0.33	0.98	0.41	0.28	0.33	0.98	0.41	0.28	0.33	0.98

prediction results can be obtained intuitively, as shown in Figure 5(c). Then, real values and predicted values of time and state will be passed into the classifier of the anomaly detection, respectively. For quantitative evaluation of forecast results, we use the normalized root mean square error (NRMSE) [28] as an indicator. For a given sequence $\mathbf{Y}$ and its estimate $\widehat{\mathbf{Y}}$, there are $\mathrm{MSE} = 1/n\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2$, and $\mathrm{NRMSE} = \mathrm{RMSE}/Y_{\max} - Y_{\min} = \sqrt{\mathrm{MSE}}/Y_{\max} - Y_{\min}$.
NRMSE relates the root mean square error (RMSE) to the observed variable range. Thus, it can be interpreted as a fraction of the overall range that is typically resolved by the model. The NRMSE of prediction results is shown in Table 3. The NRMSE for binary states prediction and timestamps prediction is 0 and 0.02, respectively. The effectiveness of the prediction achieves as expected.

5.3. Effectiveness on Anomaly Detection. First, we take an experiment on the motion sensor *M001*. We extract a total of 24,765 log entries generated by *M001* within a week. Further, we simulate 2,500 injection attacks conducted on our testbed platform. After that, temporal context features and environmental context features at the corresponding moment are extracted for anomaly detection. We select the first 80% of the data as the training set and the last 20% as the test set. We use Youden *J* statistic [29] to obtain the optimal receiver operating characteristic (ROC) threshold $\Theta = \mathrm{argmax}(\mathrm{TPR} - \mathrm{FPR}) = \mathrm{argmax}((\mathrm{TP}/\mathrm{TP} + \mathrm{FN}) + (\mathrm{TN}/\mathrm{TN} + \mathrm{FP}) - 1)$. The F1-score of abnormal events generated by the classifier is calculated as 0.90636, with an accuracy of 0.96.

To achieve better classification while avoiding overcomplicated schemes, we fine-tune the NN structure. Specifically, there are two nodes in the output layer. One acts as the normal label $\widehat{y}_1$, the other acts as the abnormal label $\widehat{y}_2$, and both fall in the range 0 to 1 through the softmax function and add up to 1, i.e., $\widehat{\mathbf{y}} = (\widehat{y}_1; \widehat{y}_2) = \mathrm{Sigmoid}(\mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3)$. During training, normal events are labeled as (1,0), and abnormal events are labeled as (0,1). During classification, once the output shows $\widehat{y}_1 > \widehat{y}_2$, the input event is classified as normal, and $\widehat{y}_1 < \widehat{y}_2$ is the criteria to determine abnormal events. We feed the dataset into a new dual-output classification network, and its results compared to the original one are shown in Table 4. F1-score is selected as the effectiveness evaluation criterion. The same methods are applied to motion sensor M002, temperature sensors T102/T103, smart lights L001/L002, and light sensors LS001/LS002. Anomaly detection of different devices has different dependencies on temporal or environmental contexts. Combining the two features can adapt to different types of devices without obvious F1-score degradation. Besides, the classifier achieves a better classification of smart devices of different value types. Since a dual-output neural network structure avoids threshold selection, it has better classification ability than a single-output neural network classifier.

6. Related Work

6.1. Event Sequence-Based Detection. Current studies mainly use network traffic and environmental sensor states at the time of the event as features. Considering network traffic features, Saxena et al. [30] propose a method to detect the identity and behavior of home devices using encrypted network traffic, choosing statistical features of ZigBee network traffic packets as a basis for classification. Zhang et al. [12] present an approach based on physical event fingerprints. They construct automata for IoT application behavior and extract event features from the wireless communication environment as fingerprints.

Since smart home sensors may change correlatively when they are affected by the same physical event, Laput et al. [31] propose a method to obtain event fingerprints based on heterogeneous sensor data. They collect data from all sensors except cameras, manually label event data, and use an SVM model to classify abnormal events. Birnbach et al. [32] also use heterogeneous sensor data to build fingerprints for events and detect spoofed events. They extract the relative mutual information of each sensor and event as fingerprints and select data for SVM classification.

6.2. Application Analysis-based Detection. In addition to extracting the characteristics of events, the correlations of IoT applications are also considered in anomaly detection.

To find risky physical channel associations, Ding et al. [11] provide IoTMon, a solution for identifying and analyzing hidden interaction chains between IoT applications. It analyzes SmartApp interaction using static analysis and SmartApp descriptions via NLP to identify smart home environments. Soteria [33] models IoT applications based on intermediate representation (IR). The state model is automatically extracted from the SmartThings IoT applications to detect whether the program has rule conflicts.

However, approaches based on static analysis cannot solve the runtime policy violation problem in realistic smart home systems. IoTGuard [34] is a dynamic policy

enforcement system that blocks insecure states by monitoring the runtime behavior of IoT applications. HAWatcher [14] is based on semantic analysis of event logs and physical-channel-related descriptions. It generates associations and uses event logs for verification. Based on the analysis of the impact of physical channels between devices, Ozmen et al. [35] use formulas of physical laws for the first time to quantify the specific results of interactions between devices, improving the accuracy of the analysis results.

7. Conclusion

In this paper, we propose HomeGuardian, a context-based approach to detect abnormal events in smart home systems. In our approach, we predict the temporal context and infer environmental context based on system log, device, and rule configurations. By converging these hybrid event contexts, we construct a learning-based classifier to detect abnormal events. We evaluate HomeGuardian based on the event data collected from our self-configured testbed. The experiment results show that the F1-scores are beyond 0.90 for all device types.

Data Availability

The public data used to support this study are available at DOI:10.1109/JBHI.2015.2461659. The prior study (and dataset) is cited at relevant places within the text.

Conflicts of Interest

The authors declare that they have no interest conflicts.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 62172027), the Beijing Natural Science Foundation (No. 4202036), the National Key R&D Program of China (No. 2020YFB1005601), Hangzhou Innovation Institute, Beihang University, under Grant 2020-Y5-A-022, and the National Natural Science Foundation of China (No. U1733115, No. 61871023).

References

[1] S. Suresh and P. V. Sruthi, "A review on smart home technology," in *2015 online international conference on green engineering and technologies (IC-GET)*, pp. 1–3, IEEE, 2015.

[2] Z. Tong, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[3] Z. Cai, Z. Xu, J. Wang, and Z. He, "Private data trading towards range counting queries in internet of things," *IEEE Transactions on Mobile Computing*, vol. 1, 2022.

[4] B. Zhang, Y. Song, and Z. Tang, "Data mining-based smart home control platform," *Office Automation*, vol. 423, no. 10, pp. 24–28, 2020.

[5] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[6] Y. Huo, J. Fan, Y. Wen, and R. Li, "A cross-layer cooperative jamming scheme for social internet of things," *Tsinghua Science and Technology*, vol. 26, no. 4, pp. 523–535, 2021.

[7] A. Acar, H. Fereidooni, T. Abera et al., "Peek-a-boo: I see your smart home activities, even encrypted!," *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 207–218, 2020.

[8] Z. Cai and Z. He, "Trading private range counting over big iot data," *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, 2019.

[9] NIST, "Cve-2018-3911 detail," https://nvd.nist.gov/vuln/detail/CVE-2018-3911, 2018.

[10] Y. Tian, N. Zhang, Y. H. Lin et al., "Smartauth:user-centered authorization for the internet of things," *26th USENIX Security Symposium (USENIX Security 17)*, pp. 361–378, 2017.

[11] W. Ding and H. Hongxin, "On the safety of iot device physical interaction control," *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 832–846, 2018.

[12] W. Zhang, Y. Meng, Y. Liu, X. Zhang, Y. Zhang, and H. Zhu, "Homonit: monitoring smart home apps from encrypted traffic," *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1074–1088, 2018.

[13] Y. Yonghua, C. Li, M. A. Jonas et al., "Detecting abnormal behaviors in smart home," in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*, pp. 37–42, IEEE, 2019.

[14] F. Chenglong, Q. Zeng, and X. Du, "Hawatcher:semantics-aware anomaly detection for appified smart homes," *30th USENIX Security Symposium (USENIX Security 21)*, pp. 4223–4240, 2021.

[15] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks: a survey towards private and secure applications," *ACM Computing Surveys*, vol. 1, 2021.

[16] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "Sok: security evaluation of home-based iot deployments," in *2019 IEEE symposium on security and privacy (sp)*, pp. 1362–1380, IEEE, 2019.

[17] U. Satapathy, B. K. Mohanta, D. Jena, and S. Sobhanayak, "An ecc based lightweight authentication protocol for mobile phone in smart home," in *2018 IEEE 13th international conference on industrial and information systems (ICIIS)*, pp. 303–308, IEEE, 2018.

[18] G. Ho, D. Leung, P. Mishra et al., "Smart locks: lessons for securing commodity internet of things devices," *Proceedings of the 11th ACM on Asia conference on computer and communications security*, pp. 461–472, 2016.

[19] T. Melamed, "An active man-in-the-middle attack on bluetooth smart devices," *Safety and Security Studies*, vol. 15, p. 2018, 2018.

[20] A. Ranieri, D. Caputo, L. Verderame, A. Merlo, and L. Caviglione, Eds., "Deep adversarial learning on google home devices," https://arxiv.org/abs/2102.13023, 2021.

[21] Y. J. Jia, Q. A. Chen, S. Wang et al., "Contexlot: Towards providing contextual integrity to appified iot platforms," in *NDSS, volume 2*, p. 2, San Diego, 2017.

[22] X. Hao, G. Zhang, and S. Ma, "Deep learning," *International Journal of Semantic Computing*, vol. 10, no. 3, pp. 417–439, 2016.

[23] PyTorch, "Bceloss," https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html, 2019.

[24] Wikipedia, "Unix time," https://en.wikipedia.org/wiki/Unix_time, 2022.

[25] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing*, vol. 26, 2013.

[26] Google, "Googlenews-vectorsnegative300," https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing, 2013.

[27] D. J. Cook, M. Schmitter-Edgecombe, and P. Dawadi, "Analyzing activity behavior and movement in a naturalistic environment using smart home techniques," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 6, pp. 1882–1892, 2015.

[28] M. V. Shcherbakov, A. Brebels, N. L. Shcherbakova, A. P. Tyukov, T. A. Janovsky, and V. A. E. Kamaev, "A survey of forecast error measures," *World Applied Sciences Journal*, vol. 24, no. 24, pp. 171–176, 2013.

[29] W. J. Youden, "Index for rating diagnostic tests," *Cancer*, vol. 3, no. 1, pp. 32–35, 1950.

[30] U. Saxena, J. S. Sodhi, and Y. Singh, "Analysis of security attacks in a smart home networks," in *2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*, pp. 431–436, IEEE, 2017.

[31] G. Laput, Y. Zhang, and C. Harrison, "Synthetic sensors: towards general-purpose sensing," *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 3986–3999, 2017.

[32] S. Birnbach and S. Eberz, *Peeves: Physical Event Verification in Smart Homes*, 2019.

[33] Z. B. Celik, P. McDaniel, and G. Tan, "Soteria: automated iot safety and security analysis," *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pp. 147–158, 2018.

[34] Z. Berkay Celik, G. Tan, and P. D. Mc-Daniel, "Iotguard: dynamic enforcement of security and safety policy in commodity iot," *NDSS*, 2019.

[35] M. O. Ozmen, X. Li, A. C. A. Chu, Z. B. Celik, B. Hoxha, and X. Zhang, "Discovering physical interaction vulnerabilities in iot deployments," https://arxiv.org/abs/2102.01812, 2021.