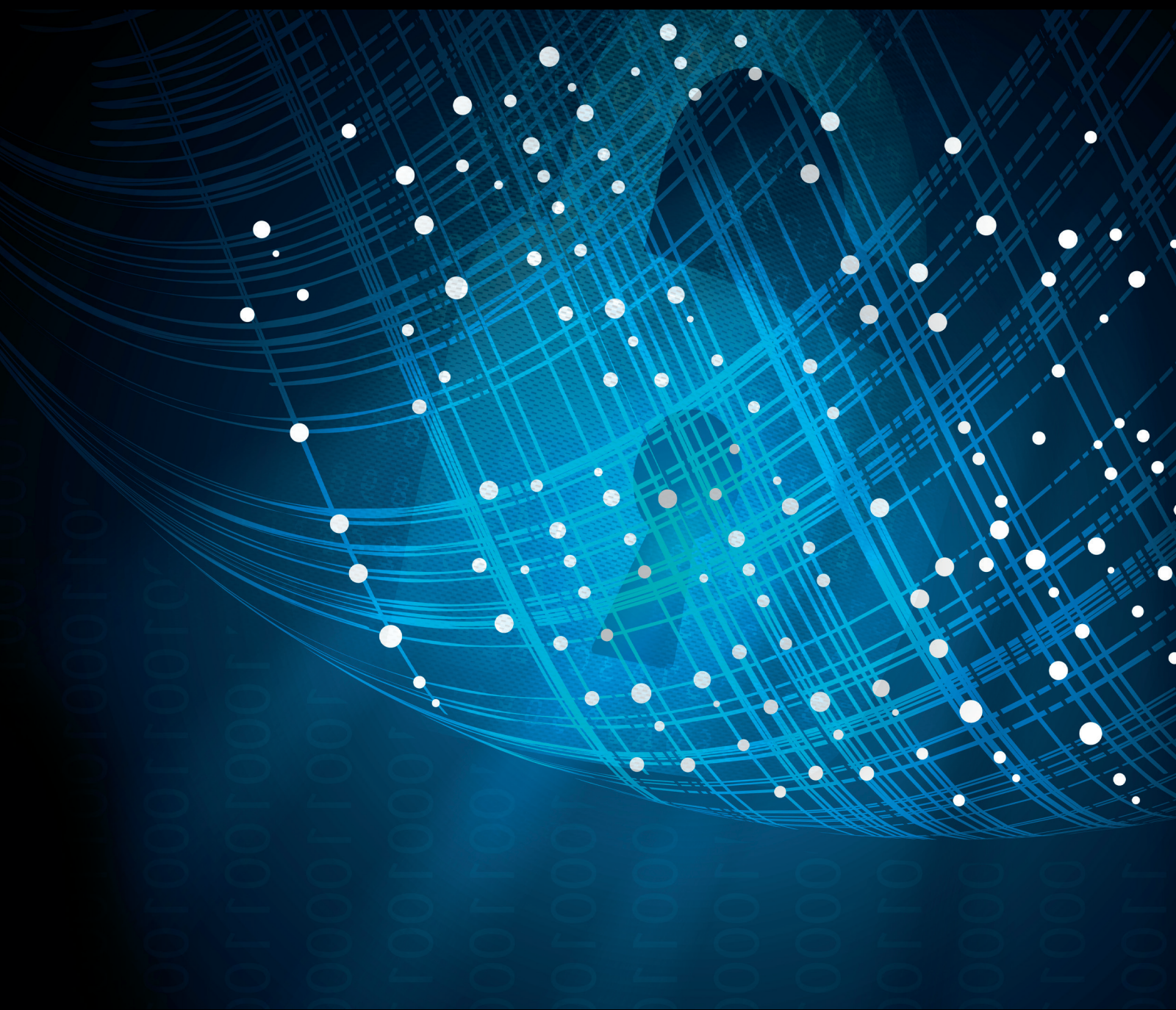
Security and Communication Networks

Security and Privacy for Smart Cyber-Physical Systems

Lead Guest Editor: Liran Ma
Guest Editors: Yan Huo, Chunqiang Hu, and Wei Li



Security and Privacy for Smart Cyber-Physical Systems

Security and Communication Networks

Security and Privacy for Smart Cyber-Physical Systems

Lead Guest Editor: Liran Ma
Guest Editors: Yan Huo, Chunqiang Hu, and Wei Li

Editorial Board

Contents

Hindawi
Security and Communication Networks
Volume 2019, Article ID 7045862, 2 pages
https://doi.org/10.1155/2019/7045862

Editorial

Security and Privacy for Smart Cyber-Physical Systems

Liran Ma,[1] **Yan Huo**,[2] **Chunqiang Hu**,[3] **and Wei Li**[4]

[1]*Texas Christian University, USA*
[2]*Beijing Jiaotong University, China*
[3]*Chongqing University, China*
[4]*Georgia State University, USA*

Correspondence should be addressed to Liran Ma; l.ma@tcu.edu

Received 17 December 2018; Accepted 18 December 2018; Published 8 January 2019

Smart cyber-physical systems (CPSs) include Internet of things (IoT), smart grids, smart cities, smart transportation, and smart "Anything" (e.g., homes and hospitals). These systems require different levels of security and protection based the sensitivity of their data. Nonetheless, we are living in a world where cyber attacks, privacy violations, phishing scams, and data breaches have become commonplace. Smart CPSs are also subject to security violations and privacy breaches, which stem from the vulnerabilities of existing computers and communications technologies. In addition, as smart CPSs get more complex, more vulnerabilities will emerge. Hackers will be able to launch increasingly sophisticated attacks in the future due to the ever-shifting cyber physical landscape. Hence, innovative research is needed for security assurance and privacy preservation in smart CPSs for new architectural models, system designs, and cryptographical protocols.

In this special issue, we received 29 submissions from both academia and industry in the relevant fields. Following a strict review process, we accepted 12 papers for this special issue. Each of the papers was peer-reviewed by at least three experts in the field. In the following, we provide a brief introduction to each paper.

There are four papers aiming to design and analyze security schemes and privacy preserving strategies for IoT applications. The paper titled "Function-Aware Anomaly Detection Based on Wavelet Neural Network for Industrial Control Communication" proposed a function-aware anomaly detection approach to detect these cyber intrusions and anomalies. Next, the authors of the paper titled "A Compatible OpenFlow Platform for Enabling Security Enhancement in SDN" proposed a unified, lightweight platform, called the open security-enhanced compatible openFlow platform, to enhance the security property and facilitate the security configuration and evaluation. And in the paper of "User Presence Inference via Encrypted Traffic of Wireless Camera in Smart Homes", the authors proposed a system (HomeSpy) that infers user presence by inspecting the intrinsic pattern of the wireless camera traffic. This system revealed that attackers are able to infer whether users are at home or not by eavesdropping the traffic of wireless cameras. Finally, in the paper titled "An Effective Integrity Verification Scheme of Cloud Data Based on BLS Signature", the authors improved the previous privacy preserving model and proposed an effective integrity verification scheme of cloud data based on Boneh-Lynn-Shacham signature. This scheme can ensure public audition and data privacy preserving.

Intelligent transportation systems, also known as vehicular ad hoc networks (VANETs), are an extended application for cyber-physical systems. Privacy preserving is also one of key issues of VANETs. Automotive intelligence is built on the dynamic data collection and application of vehicles. Vehicle driving data, collected without owners' knowledge, may be regarded as a big data gold mine so as to sale to third parties. This special issue included three papers in this topic. The paper titled "Towards a Novel Trust-Based Multicast Routing for VANETs" designed a novel trust-based multicast routing protocol to defend against multiple attacks and improve the routing efficiency. And in the work "Research on Trajectory Data Releasing Method via Differential Privacy Based on Spatial Partition", the authors discussed disclosure of trajectory data and exploited differential privacy to publish

and release trajectory data so as to ensure privacy and availability. In addition, because vehicles in VANETs can be treated as edge nodes, the authors of the paper titled "A Novel Differential Game Model-Based Intrusion Response Strategy in Fog Computing" studied the optimal intrusion response strategy and formulated a mathematical model based on differential game theory.

There are five papers focusing on privacy preserving in mobile social networks (MSNs). In the paper titled "Differentially Private Recommendation System Based on Community Detection in Social Network Applications", the authors proposed a novel recommendation method that provided a list of recommendations for target attributes based on community detection and known user attributes and links. Secondly, the work in the paper titled "RPAR: Location Privacy Preserving via Repartitioning Anonymous Region in Mobile Social Network" investigated the problem of location privacy preserving in MSNs and minimized the traffic between the anonymous server and the LBS server while protecting the privacy of the user location. Thirdly, the authors of the paper titled "Privacy Preservation for Friend-Recommendation Applications" analyzed the privacy leakage of friend-recommendation applications in social networks and put forward a privacy protection mechanism based on zero knowledge without any privacy leakage to the application server. Fourthly, different from adding noises into the original data for privacy protection, the work "GANs Based Density Distribution Privacy-Preservation on Mobility Data" studied the problem of privacy-preservation of density distribution on mobility data for location-based services in MSNs. Lastly, in the paper titled "Achieving the Optimal *k*-Anonymity for Content Privacy in Interactive Cyberphysical Systems", the authors suggested using k-anonymity solutions based on two content privacy metrics to formulate the problem of achieving the optimal content privacy in interactive cyber-physical systems.

We would like to thank all the authors for their great contributions to the Special Issue of Security and Privacy for Smart Cyber-Physical Systems and thank all anonymous reviewers for their valuable comments which help the authors to further improve their papers. It is an honor for all of us to serve as Guest Editors at Security and Communication Networks.

Conflicts of Interest

The editors declare that they have no conflicts of interest regarding the publication of this Special Issue.

Liran Ma
Yan Huo
Chunqiang Hu
Wei Li

Hindawi
Security and Communication Networks
Volume 2018, Article ID 9203076, 13 pages
https://doi.org/10.1155/2018/9203076

Research Article

GANs Based Density Distribution Privacy-Preservation on Mobility Data

Dan Yin and Qing Yang

Department of Computer Science and Technology, Harbin Engineering University, Harbin, China

Correspondence should be addressed to Qing Yang; yangqing@hrbeu.edu.cn

Received 7 September 2018; Revised 15 October 2018; Accepted 19 November 2018; Published 2 December 2018

Guest Editor: Liran Ma

With the development of mobile devices and GPS, plenty of Location-based Services (LBSs) have emerged in these years. LBSs can be applied in a variety of contexts, such as health, entertainment, and personal life. The location based data that contains significant personal information is released for analysing and mining. The privacy information of users can be attacked from the published data. In this paper, we investigate the problem of privacy-preservation of density distribution on mobility data. Different from adding noises into the original data for privacy protection, we devise the Generative Adversarial Networks (GANs) to train the generator and discriminator for generating the privacy-preserved data. We conduct extensive experiments on two real world mobile datasets. It is demonstrated that our method outperforms the differential privacy approach in both data utility and attack error.

1. Introduction

With the increasing popularity of mobile devices and GPS, plenty of Location-based Services (LBSs) have emerged in these years. LBSs can be applied in a variety of contexts, such as health, entertainment, and personal life. People can report their locations anywhere and anytime. For example, people release tweets with their current locations on social networks; users share their running routines with their friends on the Internet. The location based data which includes significant personal information is often published for analysing and mining.

The mobility data implies valuable personal information, such as home addresses, occupation, social relations, and interests. Attackers can discover the privacy information of users from the published dataset. For instance, the identities can be interred from the locations where people often visit over a period of time, even their home addresses or occupations.

In order to protect the personal information, there has been some research on the privacy-preservation of mobility data. One of them is proposed in [1], which aggregates the users in each location and publishes the aggregated results (density distribution) instead of the original location distribution. However, attackers can recover the users' mobile trajectories from the density distribution for a period of time. As shown in Figure 1, there are 3 location samples of 6 users $\{u_1, u_2, \ldots, u_6\}$ at timestamp $\{t_1, t_2, t_3\}$. The whole space is divided into 3 blocks $\{b_1, b_2, b_3\}$. If we aggregate the users in each block, we can get the density distribution from times t_1 to t_3. While this aggregate result is vulnerable to reconstruction attack [2], therefore, it is crucial to propose a new method for the density distribution privacy-preservation. As shown in the figure, we can generate similar density distribution under privacy-preservation to publish. It is hard for the adversaries to recover the users' trajectories from the published distribution.

Alternatively, there are other methods by adding noises into aggregated results for privacy-preservation. For example, differential privacy (DP) [3] is proposed aiming to provide means to maximize the accuracy of queries from statistical databases while minimizing the chances of identifying its records. However, the utility of the privacy-preserved data produced by DP method becomes worse significantly.

GANs are a class of algorithms in unsupervised machine learning, which have been widely used to produce samples of photo realistic images for the purposes of visualizing new interior design. Motivated by this, we try to utilize GANs

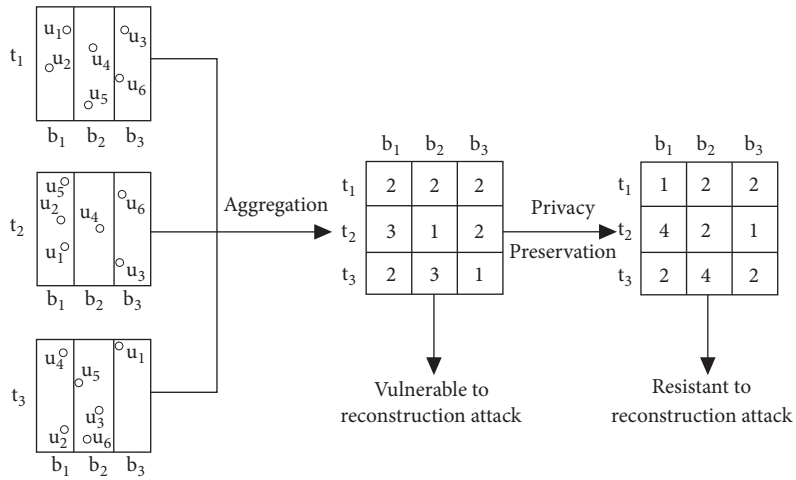


Figure 1: Density distribution privacy-preservation.

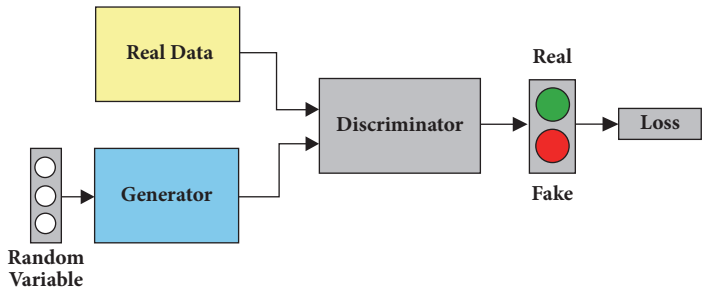


Figure 2: Architecture of GANs.

for generating privacy-preserving data with high utility. In this paper, we propose the density distribution privacy-preservation on mobility data based on GANs. By training the two neural networks, we can generate the privacy-preserved data which can achieve high data utility and low attack error. The main contributions of this paper are as follows.

(i) We investigate the privacy-preservation of density distribution on mobility data against the aggregation attack. Different from adding noises to the original data, we propose a deep learning method based on GANs to solve the problem.

(ii) Motivated by the applications of GANs on image processing, we train the generator and discriminator in GANs by random data and the original data and publish the data generated by the generator instead of the original data. To the best of our knowledge, this is the piece of paper employing GANs on data privacy-preservation.

(iii) We conduct extensive experiments on two real world datasets. The experimental results demonstrate that our method outperforms the differential privacy in both data utility and attack error.

The rest of this paper is organised as follows. Section 2 introduces the preliminaries. Section 3 introduces the proposed methods. Section 4 presents the experiment results. Section 5 describes the related work. Section 6 concludes the whole paper.

2. Preliminary

In this section, we start with the introduction of GANs, which is the basic architecture of our method. Then we describe the recently proposed attack model that recovers individual users' trajectries from density distributions, which will be adopted to measure the privacy-preservation ability of our method.

2.1. Generative Adversarial Networks. Generative Adversarial Networks (GANs) are a class of artificial intelligence algorithms used in unsupervised machine learning, which are composed of two neural networks contesting with each other in a zero-sum game framework. GANs were introduced by lan Goodfellow et al. [4] in 2014 as a novel way to model data distributions. The architecture of the general GANs is shown in Figure 2.

Specifically, the two neural networks are a generator G and a discriminator D. In the original GANs, the generator G accepts a random distribution P_z and generates synthetic data from P_z. While the goal of the discriminator D is to distinguish the synthetic data generated by G from real data $\boldsymbol{x}$, the optimal D would distinguish synthetic data from real data exactly. While the optimal G would generate synthetic

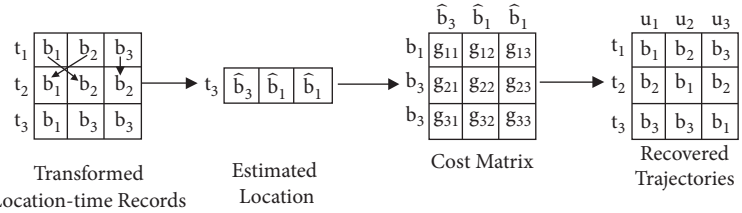


Figure 3: Procedure of reconstruction attack on transformed mobility dataset.

data that are indistinguishable from the real data for D, in the training phase of the original GANs, both the generator G and the discriminator D are iteratively optimized against each other in a minimax game with the value function $V(G, D)$, which can be formalized as

$$\min_{\theta_G} \max_{\theta_D} V(G, D) = \mathbb{E}_{x \sim P_{data}} [\log D(x)] + \mathbb{E}_{z \sim P_z} [\log (1 - D(G(z)))], \quad (1)$$

where θ_G and θ_D are the parameters of G and D, respectively.

To handle the privacy risks of releasing density distributions datasets, we generate the synthetic distributions with GANs as a protected version of the original data.

2.2. Attack Model. Many researches have been done on inference of sensitive information [5]. In order to protect the location information, some researchers aggregate the users in the same locations over a period of time and then publish the density distributions [6]. However, Fengli et al. [2] propose that releasing aggregated results does not preserve users' privacy, since a user's mobility pattern is regular while different from others'. Based on the characteristics of human mobility, they transform the density distribution to a *location-time* format and propose the trajectories reconstruction attack that iteratively associates the same users' mobility records in the neighbouring time slots. They exploit the regularity of mobility data to estimate the next location of the user and choose the location in the transformed dataset with the largest similarity to the estimated next location as the reconstructed next location according to the uniqueness pattern of human mobility data.

To recover trajectories from the density distributions, the first step is to transform the density distribution $P^t = [p_1^t, p_2^t, \ldots, p_i^t, \ldots, p_n^t]$ into a *location-time* record $B^t = [b_1^t, b_2^t, \ldots, b_j^t, \ldots, b_m^t]$, where p_i^t represents the number of users at location i during time slot t, b_j^t represents the location of the j^{th} user at time slot t, n represents the total number of possible locations, and m is the total number of users. To link the *location-time* records that represent the same users across different time slots, the reconstruction attack is modeled as a *Linear Sum Assignment Problem* [7], which can be solved in polynomial time based on *Hungarian algorithm* [8].

Specifically, we assume a set of recovered trajectories until time slot t as $S^t = [s_1^t, s_2^t, \ldots, s_k^t, \ldots, s_m^t]$, where $s_k^t = [b_k^1, b_k^2, \ldots, b_k^t]$ is the k^{th} recovered trajectory and b_k^t is the recovered location at time slot t. For the adversaries, with the abundant kinds of social networks, such as WeChat and MoMo, it is effortless to get some background information of the individuals, such as trajectories in a shot time. To recover the next position b_k^{t+1} from the *location-time* records $B^{t+1} = [b_1^{t+1}, b_2^{t+1}, \ldots, b_m^{t+1}]$, an estimated location $\widehat{b}_k^{t+1}$ is first generated based on the continuity feature of human mobility, and then the location in the *location-time* record B^{t+1} with the largest likelihood to the estimated location $\widehat{b}_k^{t+1}$ will be chosen as the recovered next location, i.e., b_k^{t+1}. In the daytime, users move frequently, and their locations are continuous, which makes it possible to estimate the next location with the current location and the velocity. Formally, for the $k^{th}(1 \leq k \leq m)$ recovered trajectory, the estimated location is

$$\widehat{b}_k^{t+1} = b_k^t + \left(b_k^t - b_k^{t-1}\right). \quad (2)$$

To quantify the likelihood between the estimated location and those in the *location-time* records, Fengli et al. [2] formulate the cost matrix $G^t = \{g_{i,j}^t\}_{m \times m}$, where $g_{i,j}^t$ is the distance between the estimated next location $\widehat{b}_i^{t+1}$ and the actual location $_j^{t+1}$.

Figure 3 presents the process of recovering the trajectories. There are three possible locations and three time slots. We assume that trajectories until time slot t_2 have been recovered, and then the estimations of the t_3 locations are generated based on the continuity feature of mobility data. The distance between the estimated locations and those in *location-time* records is formulated as the cost matrix. In the last step, *Hungarian algorithm* is applied to minimize the cost matrix and find each trajectory's associated location in the *location-time* record. The right part in Figure 3 demonstrates the recovered trajectories.

Generally, the adversaries may have different kinds of background knowledge based on various sources such as social networks. However, in our specific attack model, to ease the presentation, we assume that the adversary has the target users' location information in the first two time slots as the background knowledge.

3. Method

In this section, we first give an overview of our proposed method using GANs to generate private-preserving density distributions. Then, we describe the architecture of discriminator and generator network, respectively. Finally, we introduce the loss function of our method.

3.1. Overview. In GANs, the generator network G accepts the random data and generates the synthetic data that are similar

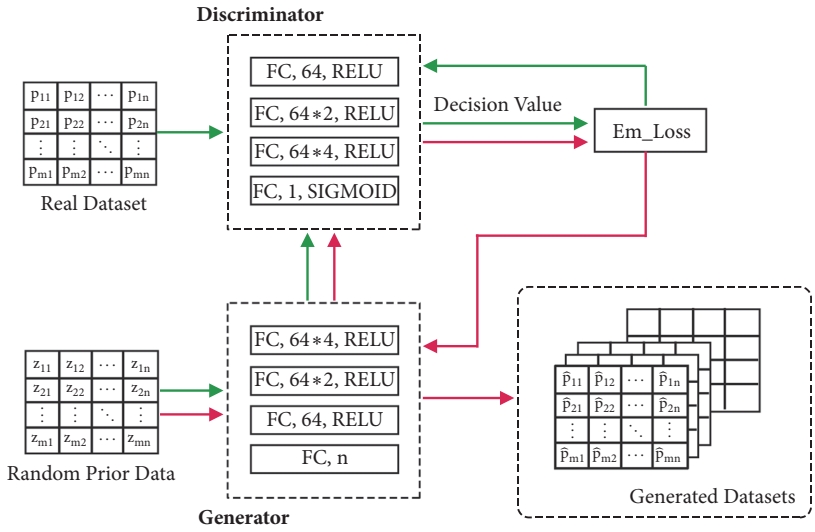


Figure 4: The training process of GANs.

Input: The real dataset P; The learning rate lr;
Output: The privacy-preserved version of the real dataset $\widehat{P}$;
1: Build discriminator network **D**;
2: Build generator network **G**;
3: $Z = randm(size(P))$;
4: **while** $lr \leq loss$ **do**
5: $D = train(D)$;
6: $G = train(G)$;
7: $loss = em\_loss(G(Z), P)$;
8: **end while**
9: $\widehat{P} = G(Z)$;

Algorithm 1: Pseudocode of the proposed privacy-preservation method based on GANs.

to the real data, while the discriminator network D tries to distinguish the generated data from the real one. At the start of the training phase of GANs, the synthetic data generated by G is easy to be distinguished by the discriminator D as G has not learned the features of the real data, which could be regarded as achieving high privacy-preserving ability, while, with the increase of the training steps, the synthetic data generated by the generator network G becomes indistinguishable for the discriminator D, as G has learned the features of the real data, which could be regarded as having less privacy-preserving ability but high utility. Based on the above observation, we propose a framework based GANs to generate privacy-preserving density distributions. We also summarize our proposed method using pseudocode in Algorithm 1.

We assume the real density distribution is $P_{n\times m}$, which is the protected target containing n blocks, i.e., $\{b_1, b_2, \ldots, b_n\}$ during m time slots, i.e., $\{t_1, t_2, \ldots, t_m\}$. We use $p_{i,j}$ to denote the number of users in block b_i at time slot t_j. To protect the real density distribution P, we build GANs to learn the potential features of P and exploit the generator network G to generate the privacy-preserving version of the real dataset that includes most of the real features and excludes the sensitive information (e.g., the individual users' mobility trajectories).

We introduce the training process of GANs in Figure 4. We show the training flow of discriminator and generator with green and red lines, respectively. When training the discriminator, we input the real data and update the parameters in D based on the loss value. Then the generated data is fed into D, which trains D to learn the features of the generated synthetic data.

When training the generator network G, we first provide a random integral matrix $Z_{n\times m}$ with the same size of the real data P. Then the synthetic data is generated by G and fed into the discriminator network D. After that, the discriminator D computes the classification of the generated data as the decision value and sends it to the loss function to compute the loss value. Based on the loss value, the generator G adjusts its parameters via backproppagation. The generated synthetic data becomes more and more similar to the real data with the number of the training rounds increasing. We train the GANs for k rounds and store the synthetic data generated by G as $\widehat{P} = \{\widehat{P_1}, \widehat{P_2}, \ldots, \widehat{P_k}\}$, where $\widehat{P_i}$ is the synthetic data generated in the i_{th} training round. The difference between the generated and real data is computed as the utility loss. With the increasing of training rounds, the utility loss experiences a downward trend via backpropogation, and the generated data becomes more and more similar to the real data. In other words, the utility of the synthetic dataset changes with each training round, and the datasets contained in $\widehat{P}$ can satisfy different utility requirements.

3.2. Discriminator Network. We design the discriminator network D with 4 layers. The first 3 layers are designed to learn the features of the input data, and the last layer is designed to compute the decision value. As the density distribution is one-dimension integral matrix, we set all the

layers in discriminator network to be fully connected with the adjacent layers and choose *Relu* as the activation function for the first 3 layers. We choose *Sigmoid* as the activation function of the last layer, so that the output is limited from 0 to 1. We set 0 as fake and 1 as real. When the input of the discriminator is the real data, the goal of the discriminator is to output a value as close to 1 as possible, and when the input is the generated data, the output should be as close to 0 as possible.

The training stage of the discriminator is composed of 3 parts. (1) Learn the features of the real data. (2) Learn the features of the generated data. (3) Learn to distinguish the real and fake data.

3.3. Generator Network. The input of the generator is a random intergal matrix Z. We set the size of Z the same as the real data. The output of the generator is the synthetic density distributions data. The generator is trained to learn the features of the real data. The generated data becomes more and more similar to the real data. In the early training stage, the generated dataset is of high privacy-preserving ability as the generator has not learned the features of the real data. With the training rounds increasing, the generated data is more similar to the real data, but less privacy-preserving.

We design the generator network G with 4 layers. In original GANs, the generator and discriminator are opponents, so we reverse the first 3 layers in the discriminator network D as the first part of the generator. Similar to the discriminator network, we fully connect all the adjacent layers and choose *Relu* as the activation function. For the last layer of the generator, we set its node number the same as the block number of the real data. All nodes are fully connected with the third layer's nodes. So the output of the generator has the same size as the real data.

We first train the discriminator D with the real data P and the generated data $\widehat{P}$ from G for a certain training rounds. Then we train the generator by combining the generator G and the discriminator D. In this stage, the parameters in D are fixed, while G adjusts its parameters based on the decision value computed by D via backpropagation. The generator's goal is to cheat the discriminator, so the generator tries to adjust it parameters that could receive the decision value from the discriminator as close to 1 (which means real data) as possible, which means the discriminator cannot distinguish the generated data from the real data.

At each training round of generator, we save the generated data $\widehat{P}$ and compute the mean square error (MSE) between $\widehat{P}$ and P as the utility loss, so that we can provide the suitable generated privacy-preserving data satisfying different utility requirements.

3.4. Loss Function. We employ the *Wasserstain distance* [9] $W(q, p)$ as the loss function, which measures the difference between the decision value and the real classification value (1 for real data and 0 for fake data). $W(q, p)$ indicates the minimum cost of transforming from the distribution q to p. The classification value of the real/generated data is $Y = [y_1, y_2, \ldots, y_m]$, where y_i is the classification value of the i_{th} record and m is the number of records in the dataset. The corresponding decision value computed by discriminator is $\widehat{Y} = [\widehat{y_1}, \widehat{y_2}, \ldots, \widehat{y_m}]$. The loss value is

$$em\_loss\left(Y, \widehat{Y}\right) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i \cdot \widehat{y_j}. \tag{3}$$

In the training phase of the discriminator, when the input is the real data, we set the classification value as 1, and when the input is the generated data, we set the classification value as 0. In the training phase of the generator, we set the classification value of the generated dataset as 1 to cheat the discriminator.

4. Evaluation

In this section, we evaluate the performance of our method. We also introduce another two privacy-preserving methods: geo-indistinguishable and exponential mechanism-based methods. Finally we compare our method with them on the trade-off between the utility loss and privacy-preservation, which is measured by the attack model.

4.1. Evaluation Metrics. In the privacy-preserving research area, the trade-off between the privacy and utility is the focus attention. In this part, we demonstrate the performance of our method on privacy-preservation under the attack model and utility loss compared with the real world datasets.

We quantify the privacy-preservation performance by the *reconstruction error* under the attack model, which is defined as the Euclidean distance between the reconstructed individual users' trajectories and the ground truth. A larger *reconstruction error* indicates that the density distributions protected by our method are not vulnerable to be attacked and our method achieves a better privacy-preservation performance.

We employ the *mean square error* (MSE) to measure the difference between the generated density distributions and the real world dataset. We quantify the *utility loss* by MSE. A smaller MSE means the density distributions generated by our method are more similar to the real datasets.

4.1.1. Reconstruction Error. The attack model introduced in Section 2.2 aims to reconstruct the individual users' trajectories from the density distributions. We compute the average Euclidean distance (reconstruction error) between the reconstructed trajectories and the ground truth to measure the privacy-preservation performance.

We assume the number of users in the real density distribution is u, and the trajectory of the $i_{th}(0 \le i \le u)$ user is $s_i = [l_i^1, l_i^2, \ldots, l_i^m]$, where m is the number of the time slots and each element is the location block of the user at a specific time slot. The corresponding reconstructed trajectory of the user is $\widehat{s_i} = [\widehat{l_1^i}, \widehat{l_2^i}, \ldots, \widehat{l_m^i}]$. We compute the reconstruction error as

$$Reconstruction\ Error = \frac{\sum_{i=1}^{u} \left\|\left(\widehat{s_i} - s_i\right)\right\|_2^2}{u}, \tag{4}$$

where $\|(\widehat{s}_i - s_i)\|_2^2$ is the Euclidean distance between the reconstructed and ground truth trajectories. A larger reconstruction error indicates the attack is unsuccessful, while the privacy is protected better.

4.1.2. Utility Loss. We measure the utility loss of protected version of the real world dataset by computing the mean square error between the real world dataset P and its privacy protected version $\widehat{P}$.

Formally, we denote the real density distributions as $P_{n\times m}$, where n is the total number of the location blocks and m is the total number of the time slots. We use $p_{i,j}(0 \le i \le n, 0 \le j \le m)$ to represent the number of users at block i, in time slot j. We denote the corresponding protected version of the real density distributions as $\widehat{P}_{n\times m}$ with the same size of the real world dataset, and $\widehat{p}_{i,j}$ represents the number of users in block i at time slot j in the protected dataset. The utility loss can be computed as

$$Utility\ Loss = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(p_{i,j} - \widehat{p}_{i,j}\right)^2}{m \times n}. \tag{5}$$

A smaller utility loss indicates the data in the protected dataset is more similar to the real dataset and better practical usability.

4.2. Compared Methods. In this part, we introduce another two privacy-preserving methods commonly used in the recent research: the geo-indistinguishability method [1] and exponential mechanism [3] based method.

Geo-Indistinguishability Method. This method is proposed by Andrés et al. to protect the location-based data with a differentially private mechanism. In this method, Laplacian noises [1] are employed to generate a radius, and the real location data is remapped by the radius with a random angle. We call the dataset protected by this method Geo-MDA.

Exponential Mechanism Based Method. The exponential mechanism [3] is one of the most renowned tools used in differential privacy. The general idea of exponential mechanism is sampling an output from the output space according to a utility function. In our experiments, we employ the proportion of users in each location block as the utility function and sample the user numbers of each location under different parameter settings. We call the dataset protected by this method Exp-MDA.

4.3. Datasets. We use two real world mobility datasets, MoMo mobile app dataset and San Francisco cabs dataset.

MoMo Mobile App Dataset (MoMo) [10]. MoMo is a mobile social networking application in China. This dataset was collected from the GPS of the mobile devices using MoMo from 21 May, 2012, to 26 June, 2012, in Beijing, China. Each record in the dataset contains the user ID, timestamp, latitude, and longitude.

San Francisco Cabs Dataset (SFC) [11]. This dataset contains the mobility trajectories of taxi cabs in San Fancisco, USA. This dataset was collected over 30 days in the San Francisco bay area. Each record has four attributes: cab ID, timestamp, latitude, and longitude.

After the preprocessing of the raw datasets, we choose 198 users' trajectory records from MoMo dataset and set the spatial resolution 2km and the temporal resolution 30 minutes. For the SFC dataset, we choose 127 users' trajectories and set the temporal resolution 2 minutes, as the mobility speed of human beings is much slower than the taxi cabs. The size of the area for both the datasets is $50km \times 50km$, and the location blocks number is 625. In the preprocessing stage, we transform the individual users' mobility records into the density distributions P, that is, counting the users number in each location at each time slot.

4.4. Privacy-Preserving Performance against the Attack Model. We first apply the attack model on the real density distributions datasets of MoMo and SFC. The average reconstruction errors obtained by the reconstruction attack on MoMo and SFC are 4.32km and 1km, respectively. We regard these reconstruction errors as the baseline in our evaluation experiments and represent them as horizontal lines in Figures 5 and 7.

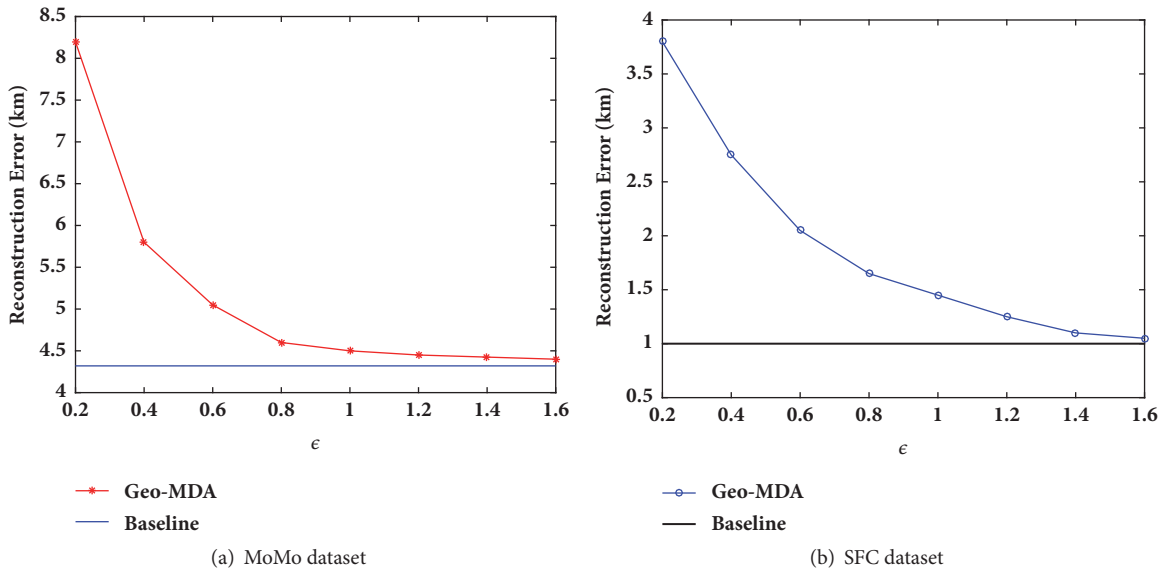
We evaluate the privacy-preservation ability of the Geo-MDA against the attack model. The results are shown in Figure 5. The x-axis shows the parameter ϵ of this method, which controls the noise level, and the y-axis stands for the reconstruction error. We evaluate the Geo-MDA with the parameter ϵ from 0.2 to 1.6. As the results show in Figure 5, for both the MoMo and SFC datasets, the reconstruction error decreases when ϵ increases, which indicates that when the value of ϵ increases, the privacy-preservation ability of Geo-MDA becomes weak.

Then we evaluate the privacy-preservation ability of the exponential mechanism method (Exp-MDA). The results are shown in Figure 6. In the exponential mechanism, we test the value of ϵ from 0; however, for MoMo dataset, when we vary ϵ from 0 to 5, the reconstruction error is stable, and for both datasets, when ϵ is larger than 13.2, the reconstruction error starts to increase. We only show the results with ϵ ranging from 5 to 13.2 for MoMo dataset, and for the SFC dataset, we show the results with ϵ ranging from 5.8 to 13.2.

Figure 6 shows that the reconstruction errors of both datasets are decreasing when ϵ value is increasing. And the minimum reconstruction error for MoMo is about 19.2 km and for SFC is 5.3 km; both are larger than the baseline of the real dataset. Besides, we observe that when the value of ϵ is around 6 and 7, for MoMo and SFC, respectively, the reconstruction error remains stable, because, in the exponential mechanism, the output changes slowly when ϵ is small, and with the increase of ϵ, the changes become quicker as shown in Figure 6. The Exp-MDA could provide protection to the real dataset.

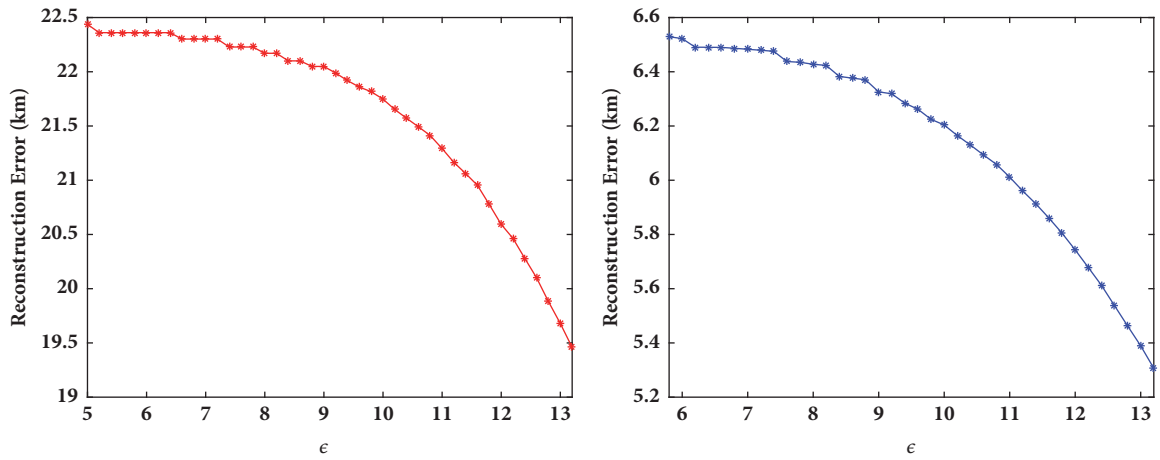
In our method, we save the generated dataset at each training time. To evaluate the privacy-preservation performance against the attack model of our method, we conduct the reconstrction attack on the fake datasets generated by each training time, and the results are shown in Figure 7.

In Figure 7, the x-axis represents the training times, and the y-axis is the corresponding reconstruction error.



(a) MoMo dataset (b) SFC dataset

Figure 5: The reconstruction error of Geo-MDA under different noise factor ϵ.



(a) MoMo dataset (b) SFC dataset

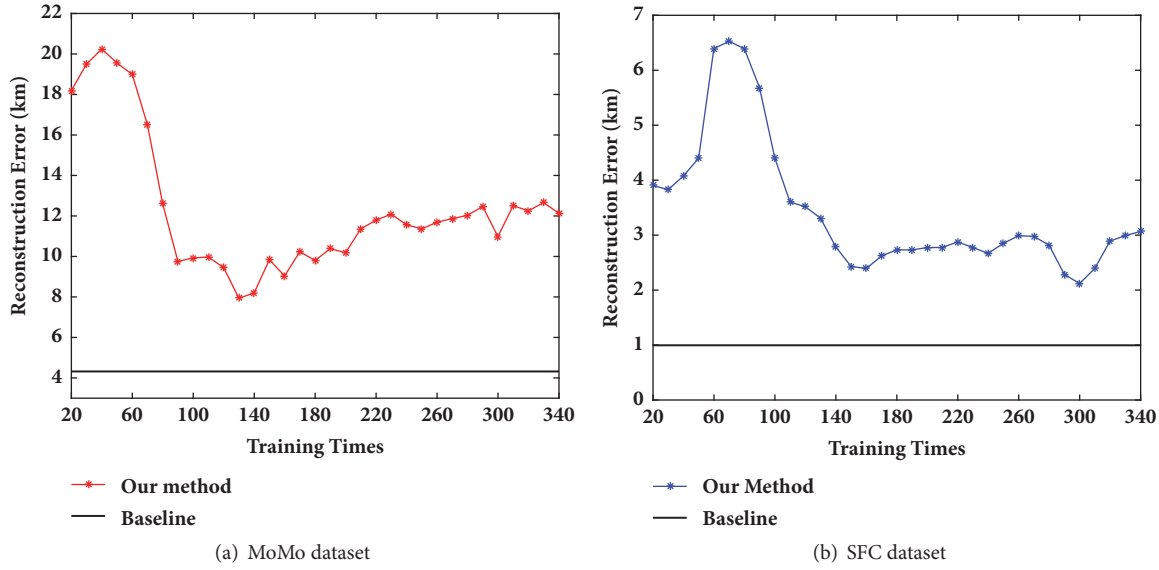
Figure 6: The reconstruction error of Exp-MDA under different noise factor ϵ.

We observe that the shape of the results is wavy; that is, because the training phase of GANs is adversarial, the generator and discriminator are trained in turn. However, the trend of the results is decreasing, which indicates that the privacy-preserving ability is decreasing with the training time growing. For MoMo dataset, the reconstruction error is limited within 10 ~ 12km (larger than the baseline 4.32km) when the GANs are trained more than 220 times. And for the SFC dataset, when the training time is larger than 180, the reconstruction error is constrained between 2km and 3km, which is also larger than the baseline value (1km).

We cannot compare our method with Geo-MDA and Exp-MDA by now, because their parameters are different, and we need to consult the utility-preservation ability of these methods as well.

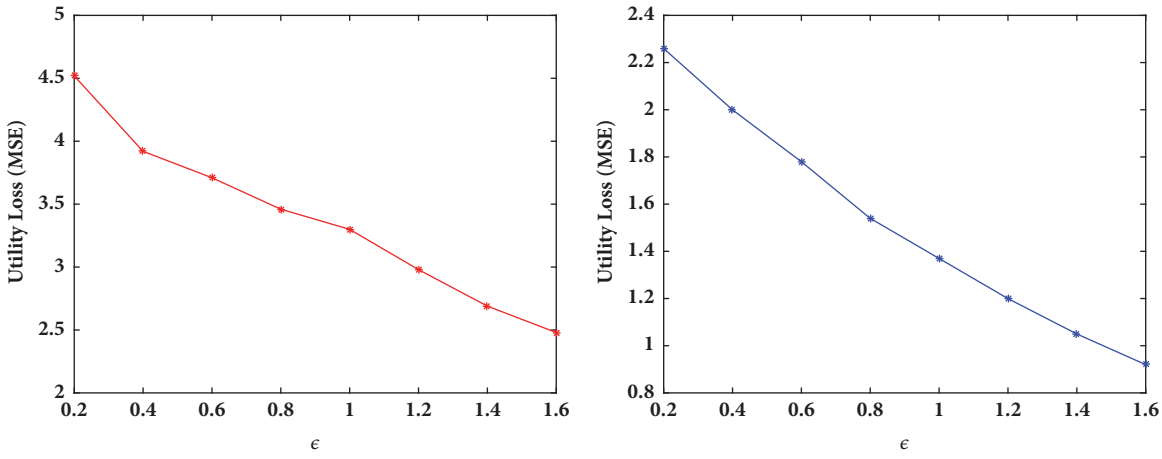
4.5. Performance on the Utility-Preservation. In this section, we evaluate the utility-preservation ability of the Geo-MDA, Exp-MDA, and our method. The utility-preservation performance is quantified by the utility loss, which has been introcduced in Section 4.1.2. A smaller utility loss indicates that the difference between the protected dataset and the real dataset is small, and the protected dataset generated by the privacy-preserving methods is of high practical usability.

The utility-preservation performance of the Geo-MDA is shown in Figure 8. The x-axis represents the parameter of Geo-MDA, ϵ, which controls the noise level of this method. And the y-axis is the utility loss. We observe that when we vary ϵ from 0.2 to 1.6, the values of the utility loss for both MoMo dataset and SFC dataset represent a downtrend. When ϵ is equal to 1.6, the utility loss for MoMo and SFC datasets



(a) MoMo dataset (b) SFC dataset

Figure 7: The reconstruction error of our method.



(a) MoMo dataset (b) SFC dataset

Figure 8: The utility loss of Geo-MDA under different noise factor ϵ.

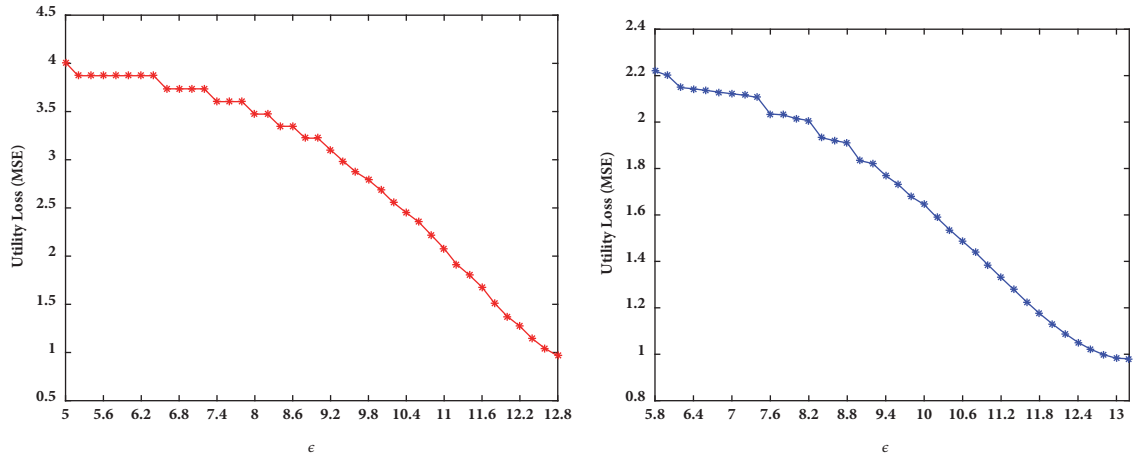
is 2.5 and 0.92, repectively, which shows that the utility-preservation ability of both MoMo and SFC is improving with the growth of ϵ.

We then evaluate the utility-preservation ability of the Exp-MDA under different parameter settings. The results are shown in Figure 9.

The x-axis is the parameter ϵ of this method, and the y-axis indicates the utility loss of the protected dataset under different parameter settings. For both the MoMo and SFC datasets, the utility loss decreases with the increase of ϵ. For MoMo dataset, when we increase ϵ from 5 to 12.8, the utility loss decreases from 4 to 1. The condition of the SFC dataset is similar to that of the MoMo dataset. Similar to the Geo-MDA, the utility-preservation ability of the Exp-MDA method improves with the increase of ϵ.

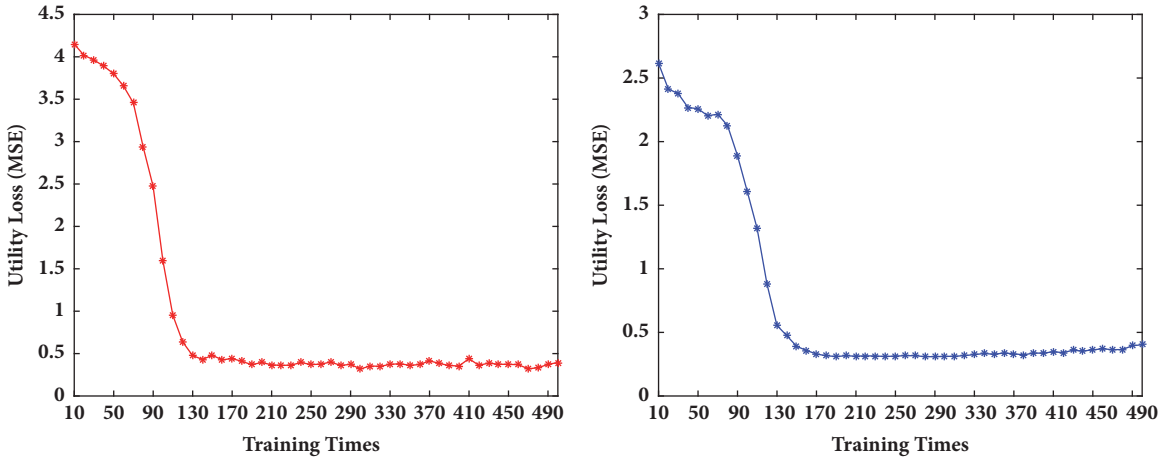
We then evaluate the utility loss that existed in our method under different training times. The results are shown in Figure 10. For both MoMo and SFC datasets, we train the GANs 500 times and save the generated datasets every 10 training times. We observe that as the training times increase, the utility loss decreases and becomes stable. For the MoMo dataset, the utility loss stays about 0.4 when the training times are larger than 130. On the other side, the utility-preservation ability will not increase when the train times are larger than 130 and 170 for MoMo and SFC datasets, respectively.

In Figure 11, we present some actual examples protected by our method to illustrate the utility loss value and the protected dataset in practice. We use colours with different gradations to show different numbers of users on the map, and the lighter the colour is, the larger the number of users in that location is. Straightforwardly, we can observe that when the utility loss is 2.25, the practical usage of the dataset generated by our method is weak and when the utility loss decreases to 0.4, the dataset generated by our method is very



(a) MoMo dataset (b) SFC dataset

Figure 9: The utility loss of Exp-MDA under different noise factor ϵ.



(a) MoMo dataset (b) SFC dataset

Figure 10: The utility loss of our method.

similar to the real dataset, which achieves better practical usage.

4.6. Privacy-Utility Trade-Off Comparison. We compare the performance of our method with the Geo-MDA and Exp-MDA by the privacy-utility trade-off in this part. The comparison is conducted by combining the privacy-preservation evaluation and the utility-preservation evaluation. The results are shown in Figure 12.

The x-axis represents the reconstruction error, which denotes the privacy-preservation ability, and the y-axis is the utility loss. We observe that, for both the MoMo and SFC datasets, under the same reconstruction error, the utlity loss of our method is smaller than the other two methods. For example, in Figure 12(b), for the SFC dataset, when the reconstruction error is 2.8km, the utility loss of the Geo-MDA and the Exp-MDA is 2 and 1.5, respectively, while the utility loss of our method is only 0.5 under the same reconstruction error. Besides, in Figure 12(a), even when the reconstruction error is as high as 12.5km, our method still preserves the utility loss less than 0.5.

5. Related Work

In this section, we start with the introduction about the services of the mobility dataset. Then we present the applications of GANs. Finally, we summarize the existing privacy-preserving methods for mobility datasets releasing.

5.1. Services for the Mobility Datasets. With the fast development of the mobile smart devices and the Internet technology, a huge number of services are developed based on the mobility datasets to provide useful information to the users. These services support human daily life by studying mobility patterns from trillions of trails and footprints [12]. Urban planning [13], face recognition [14], classification [15], traffic forecasting [16], marker campaign [17], prediction of epidemics [18, 19], latent data privacy [20], and designing

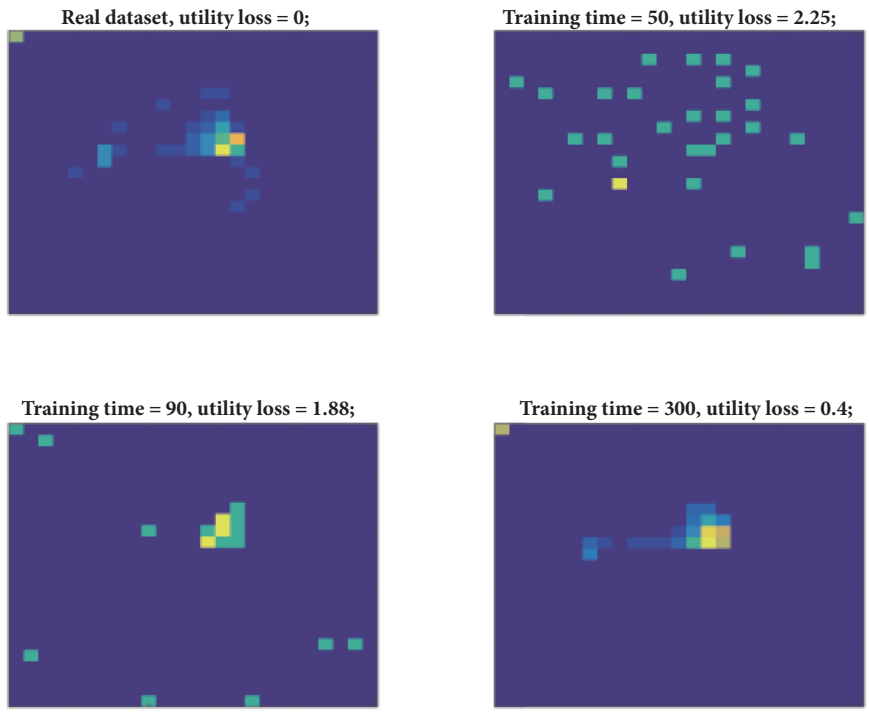
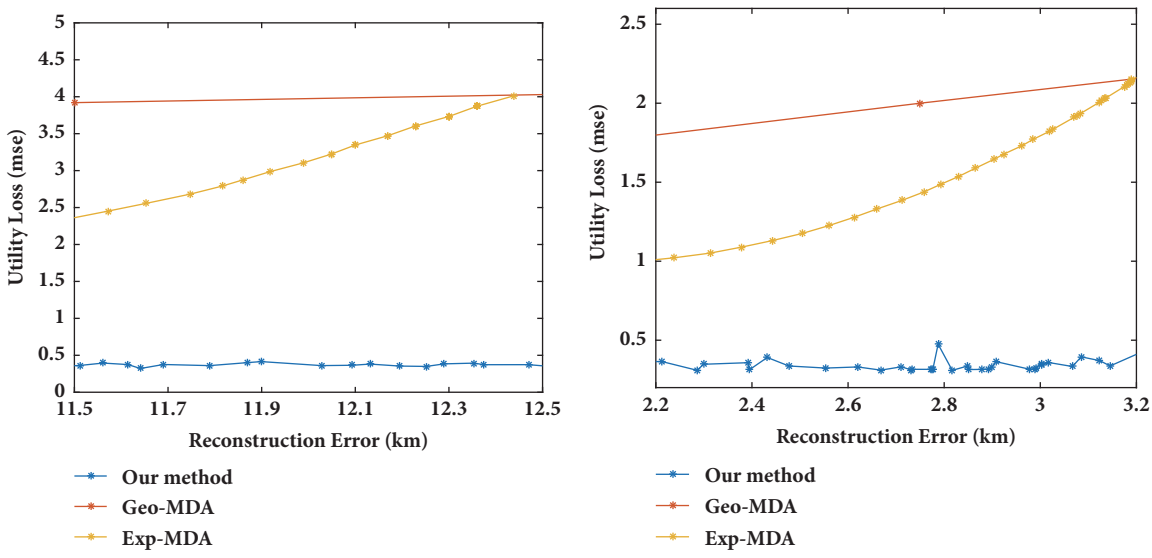


Figure 11: Actual examples of our method and the corresponding utility loss.



(a) MoMo dataset (b) SFC dataset

Figure 12: The privacy-utility trade-off comparison.

of mobile network protocols [21] are all powered by human mobility trajectories. Some other services exploit the users' daily mobility datasets finding the mobility patterns and mining users' activities to provide useful extensive services [22]. Gonzlez et al. [23] discovered that the human mobility has spatiotemporal regularity, which indicates people are very likely to return to a few frequently visited locations. However these services [24] endanger users' security and privacy as the datasets mining algorithms can link the mobility datasets to a variety of sensitive information as studies in [25, 26].

5.2. Applications of Generative Adversarial Networks (GANs). Researches on GANs are mainly in two directions. One direction is about the variants of GANs, which is aiming to solve the problems of the original GANs. For example, WGANs [27, 28] and DCGANs [29] are proposed to improve the stability of training and to alleviate mode collapse.

The other research direction of GANs focuses on the applications of GANs, and most of such researches are in the area of image processing. Radford et al. [29] use convolutional neural networks to improve image processing capacity.

The authors in [30–32] design the GANs as a conditional architecture to generate higher quality images. Reed et al. [33] combine GANs with the natural language processing technology and propose the text-to-image generation.

5.3. Privacy-Preserving Methods for Mobility Datasets Releasing. Researches on privacy-preserving mobility datasets releasing [34, 35] are becoming popular, as the mobility datasets contain sensitive individual information [36–39]. One popular method in solving the privacy issue is releasing the statistics of the mobility datasets instead of the individual trajectories. For example, the French XData project [6] only reports the density of each region in the area, which could conceal the individual information. However, the recently proposed attack method in [2] shows that such aggregation method is not safe, and they propose an approach that could recover the individual trajectories from such aggregation by exploiting the uniqueness and regularity of human mobility.

Encrypting or encoding the mobility datasets before releasing is another research direction to protect the datasets. In [1], the authors add Laplacian noise to the data, which achieves ϵ-differential privacy. There also have been many researches following other principles [40]: position dummies [41], rumor spreading [42], data aggregation [43], spatial obfuscation [44], coordinate transformation [45], and position sharing [46]. Approaches following these principles could provide privacy protection, but the trade-off between the privacy-preservation ability and the utility loss is another main focus for them.

6. Conclusions

This paper investigates the density distribution privacy-preservation on mobility data. We design a deep learning framework based on GANs. To the best of our knowledge, this is the piece of paper employing GANs on data privacy-preservation. We train the generator and discriminator in GANs by random data and the original data and publish the data generated by the generator. Adversaries cannot easily recover the users' trajectories from the published density distribution. We conduct plenty of experiments on the real world datasets. It is demonstrated that our method performs better than the compared approaches on data utility and privacy-preservation.

Data Availability

The San Francisco Cabs data have been deposited in the CRAWDAD dataset and can be downloaded from https://crawdad.org/epfl/mobility/20090224 (2009). The MoMo mobile application data are from previously reported studies and datasets. The prior study has been cited at relevant places within the text as [10].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Dan Yin and Qing Yang contributed equally to this work.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China under Grant 61702132, Natural Science Foundation of Heilongjiang province under grant QC2017071, the Fundamental Research Funds for the Central Universities Grant No. HEUCFM 180603, and the China Postdoctoral Science Foundation No. 2018M631913.

References

[1] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: differential privacy for location-based systems," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*, pp. 901–914, ACM, Berlin, Germany, November 2013.

[2] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, "Trajectory Recovery From Ash," in *Proceedings of the the 26th International Conference*, pp. 1241–1250, Perth, Australia, April 2017.

[3] C. Dwork, "Differential privacy: a survey of results," in *Theory and Applications of Models of Computation: 5th International Conference, TAMC 2008, Xi'an, China, April 25–29, 2008. Proceedings*, vol. 4978 of *Lecture Notes in Computer Science*, pp. 1–19, Springer, Berlin, Germany, 2008.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014*, pp. 2672–2680, Canada, December 2014.

[5] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[6] G. Acs and C. Castelluccia, "A case study: privacy preserving release of spatio-temporal density in Paris," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014*, pp. 1679–1688, ACM, August 2014.

[7] "Linear Programming and Extensions," *Students Quarterly Journal*, vol. 34, no. 136, p. 242, 1964.

[8] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[9] S. Vallender, "Calculation of the wasserstein distance between probability distributions on the line," *Theory of Probability & Its Applications*, vol. 18, no. 4, pp. 784–786, 1974.

[10] T. Chen, M. A. Kaafar, and R. Boreli, "The where and when of finding new friends: Analysis of a location-based social discovery network," in *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media, ICWSM 2013*, pp. 61–70, USA, July 2013.

[11] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, CRAWDAD dataset epfl/mobility (v. 2009-02-24)," Downloaded from https://crawdad.org/epfl/mobility/20090224, 2009.

[12] X. Liang, X. Zheng, W. Lv, T. Zhu, and K. Xu, "The scaling of human mobility by taxis is exponential," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 5, pp. 2135–2144, 2012.

[13] H. D. Rozenfeld, D. Rybski, J. S. Andrade Jr., M. Batty, H. E. Stanley, and H. A. Makse, "Laws of population growth," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 105, no. 48, pp. 18702–18707, 2008.

[14] W. Xu, Y. Shen, N. Bergmann, and W. Hu, "Sensor-Assisted Multi-View Face Recognition System on Smart Glass," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 197–210, 2018.

[15] Y. Shen, C. Luo, D. Yin, H. Wen, R. Daniela, and W. Hu, "Privacy-preserving sparse representation classification in cloud-enabled mobile applications," *Computer Networks*, vol. 133, pp. 59–72, 2018.

[16] B. Jiang, J. Yin, and S. Zhao, "Characterizing the human mobility pattern in a large street network," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 80, no. 2, Article ID 021136, 2009.

[17] E. Agliari, R. Burioni, D. Cassi, and F. M. Neri, "Word-of-mouth and dynamical inhomogeneous markets: an efficiency measure and optimal sampling policies for the pre-launch stage," *IMA Journal of Management Mathematics*, vol. 21, no. 1, pp. 67–83, 2010.

[18] L. Hufnagel, D. Brockmann, and T. Geisel, "Forecast and control of epidemics in a globalized world," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 101, no. 42, pp. 15124–15129, 2004.

[19] Z. He, Z. Cai, and X. Wang, "Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks," in *Proceedings of the 35th IEEE International Conference on Distributed Computing Systems (ICDCS '15)*, pp. 205–214, July 2015.

[20] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1-1, 2017.

[21] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "Slaw: A new mobility model for human walks," in *Proceedings of the INFOCOM 2009, IEEE*, pp. 855–863, 855-863. IEEE. doi, 2009.

[22] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *Proceedings of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, May 2017.

[23] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[24] X. Zheng, Z. Cai, G. Luo, L. Tian, and X. Bai, "Privacy-preserved community discovery in online social networks," *Future Generation Computer Systems*, 2018.

[25] A. LaMarca, M. Langheinrich, and K. N. Truong, *Pervasive Computing*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[26] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "De-anonymization attack on geolocated data," *Journal of Computer and System Sciences*, vol. 80, no. 8, pp. 1597–1614, 2014.

[27] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the International Conference on Machine Learning*, pp. 214–223, 2017.

[28] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

[29] A. Radford, L. Metz, and S. Chintala, *Unsupervised representation learning with deep convolutional generative adversarial networks*, 2015.

[30] C. Ledig, L. Theis, F. Huszár et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 105–114, USA, July 2017.

[31] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, p. 5, 2018.

[32] T. Karras, T. Aila, S. Laine, and J. Lehtinen, *Progressive growing of gans for improved quality, stability, and variation*, 2017.

[33] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Proceedings of the 30th Annual Conference on Neural Information Processing Systems, NIPS 2016*, pp. 217–225, Spain, December 2016.

[34] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems>," *EEE Transactions on Network Science & Engineering*, 2018.

[35] Y. Shen, H. Wen, C. Luo et al., "GaitLock: Protect Virtual and Augmented Reality Headsets Using Gait," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[36] 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 Secure Computing*, vol. 15, no. 4, p. 1, 2018.

[37] R. Wu, G. Luo, J. Shao, L. Tian, and C. Peng, "Location prediction on trajectory data: A review," *Big Data Mining and Analytics*, vol. 1, no. 2, pp. 108–127, 2018.

[38] L. Shi, Y. Wu, L. Liu, X. Sun, and L. Jiang, "Event detection and identification of influential spreaders in social media data streams," *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 34–46, 2018.

[39] 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.

[40] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 163–175, 2014.

[41] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of the 2nd International Conference on Pervasive Services (ICPS '05)*, pp. 88–97, IEEE Press, July 2005.

[42] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, "Cost-efficient strategies for restraining rumor spreading in mobile social networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2789–2800, 2017.

[43] Y. Huo, C. Yong, and Y. Lu, "Re-adp: Real-time data aggregation with adaptive w-event differential privacy for fog computing," *Wireless Communications and Mobile Computing*, pp. 1–13, 2018.

[44] C. Reynold, Y. Zhang, E. Bertino, and S. Prabhakar, "Preserving user location privacy in mobile data management infrastructures," in *Privacy Enhancing Technologies*, G. Danezis and P. Golle, Eds., vol. 4258 of *Lecture Notes in Computer Science*, pp. 393–412, Springer, Berlin, Germany, 2006.

[45] M. L. Yiu, C. S. Jensen, J. Møller, and H. Lu, "Design and analysis of a ranking approach to private location-based services," *ACM Transactions on Database Systems (TODS)*, vol. 36, no. 2, p. 10, 2011.

[46] F. Dürr, P. Skvortsov, and K. Rothermel, "Position sharing for location privacy in non-trusted systems," in *Proceedings of the 9th IEEE International Conference on Pervasive Computing and Communications, PerCom 2011*, pp. 189–196, IEEE, Seattle, Wash, USA, March 2011.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 2615249, 11 pages
https://doi.org/10.1155/2018/2615249

Research Article

An Effective Integrity Verification Scheme of Cloud Data Based on BLS Signature

Xiling Luo,[1] Zequan Zhou,[1] Lin Zhong,[1,2] Jian Mao ,[2] and Chaoyong Chen[3]

[1]*School of Electronic and Information Engineering, Beihang University, Beijing, China*
[2]*School of Cyber Science and Technology, Beihang University, Beijing, China*
[3]*The Second Research Institute of Civil Aviation Administration of China, Beijing, China*

Correspondence should be addressed to Jian Mao; maojian@buaa.edu.cn

Received 8 September 2018; Accepted 7 November 2018; Published 19 November 2018

Guest Editor: Yan Huo

Cloud storage services allow users to outsource their data remotely to save their local storage space and enable them to manage resources on demand. However, once users outsourced their data to the remote cloud platform, they lose the physical control of the data. How to ensure the integrity of outsourced data is the major concern of cloud users and also is the main challenge in the cloud service deployment. Limited by the communication and computation overheads, traditional hash-based integrity verification solutions in the stand-alone systems cannot be directly adopted in remote cloud storing environment. In this paper, we improve the previous privacy preserving model and propose an effective integrity verification scheme of cloud data based on BLS signature (EoCo), which ensures public audition and data privacy preserving. In addition, EoCo also supports batch auditing operations. We conducted theoretical analysis of our scheme, demonstrated its correctness and security properties, and evaluated the system performance as well.

1. Introduction

Recent years, cloud storage services are getting more and more popular, which allow users to outsource their data remotely to save their local storage space and enable them to manage resources on demand. However, once users outsourced the data to the remote cloud platform, they lose the physical control of their data. Some cloud service providers (CSPs) may remove users' data that are less accessed to gain more profits. In addition, the loss of data may also be caused by system crashes or operation errors [1, 2]. In this situation, CSPs may conceal the fact that users' data have no longer been stored correctly. How to ensure the integrity of outsourced data is a major concern of cloud users. Being limited by the communication and computation overheads, traditional hash-based integrity verification solutions for stand-alone systems cannot be directly adopted in remote cloud storing environments. For example, a strawman solution is to compute and keep the message authentication code of the file before outsourcing and then retrieve the whole file from the cloud server and check with the message authentication code. However, this will introduce unacceptable communication and computational overhead.

To verify the integrity of cloud data without retrieving the whole outsourced files, Ateniese et al. [3] proposed a provable data possession (PDP) scheme, which provides remote integrity verification based on homomorphic tag and sampling techniques. Their improved approach supports public audition but cannot ensure the privacy preserving property. Juels et al. [4] proposed a proof of retrievability (POR) protocol to audit and ensure the correctness of remote data once data corruption happened. Nevertheless, the scheme only supports limited rounds of verification without privacy protection of public audition. The scheme proposed by Shacham et al. [5] cannot protect data privacy either. Although many other public auditing solutions have been proposed [6–12] that preserves privacy during the integrity audition, some schemes [6, 8, 9] only prevent the data leakage, rather than provide rigorous privacy guarantee. The solutions proposed by Wang et al. and Worku et al. [6, 9] fail to achieve reliable data protection under their security models.

In this paper, we propose an effective verification of cloud data integrity scheme, EoCo, based on BLS (Boneh-Lynn-Shacham) signature with strong privacy protection. Besides efficient remote data integrity verification, our scheme also supports public auditability, privacy protection, blockless verification, and batch audition. We formally define a new integrity protection model based on the work proposed by Worku et al. [9] and prove that dishonest server cannot bypass the verification based on the CDH (computational Diffie-Hellman) problem. We also introduce a ZK-privacy (zero knowledge privacy) model and prove that our scheme is secure against CMA (adaptive chosen messages attack) attacks and CPA (chosen plaintext attack) attacks under the security assumption of the cryptographic primitives.

Contributions. Our contributions can be summarized as follows:

(i) We propose a remote data integrity protection scheme that ensures public auditability, privacy protection, and blockless verification. Besides, our scheme may also support batch auditing operations.

(ii) We introduce and formally define a ZK-privacy model, where the adversary obtains zero knowledge from the auditing interactions. We prove the privacy preserving property of our scheme under the ZK-privacy model.

(iii) We evaluate the performance of our proposed scheme through mathematically analysis and compare it with related schemes in communication and computation overhead. The communication overhead of EoCo is only $O(1)$.

Paper Organization. The rest of this paper is organized as follows. In Section 2, we discuss the related work and the key challenges; Section 3 presents the preliminaries of the proposed scheme; in Section 4, we describe the system model and the security goals; in Section 5, we propose our publicly auditable integrity verification scheme; we conduct theoretical analysis with security proof in Section 6; in Section 7, we evaluate the performance overhead of our approach; and Section 8 concludes the paper.

2. Related Works

2.1. Provable Data Possession and Proof of Retrievability. Ateniese et al. [3] first proposed a remote auditing system using a provable data possession (PDP) model to ensure data integrity in untrusted storage services. To deploy RSA (Rivest-Shamir-Adleman) digital signature, the scheme splits data into small file blocks. The high probability guarantee of data integrity is achieved by randomly selecting some blocks and check the correctness by using the attribute of RSA homomorphic linear validation. Ateniese et al. [13] introduced the retrievability property on the basis of PDP using error-correcting codes. However, as the linear combination of sample file blocks is exposed to external auditors in the public auditing process, the above two solutions cannot achieve provable privacy protection. Juels et al. [4] proposed POR model to construct data integrity verification. The scheme ensures data integrity and retrievability through spot-checking and error-correcting codes. Spot-checking is to randomly embed special check blocks, sentinels, into the data file, and then randomly select a number of sentinels to verify the file data's integrity. However, this scheme has a critical problem that once the times of validations is beyond a certain number, these fixed sentinels will be exposed and the data integrity will not be guaranteed. In addition, the scheme cannot support public audition. Bowers et al. [14] improved the POR model, and Shacham et al. [5, 15] also proposed an improved POR scheme base on BLS signature [16]. However, these schemes cannot ensure privacy protection for the same reason as the scheme proposed by Ateniese et al. [3]. Shah et al. [12, 17] introduced a third party auditor (TPA) and sent a number of precomputed symmetric-keyed hashes over the encrypted data to the TPA. Nevertheless, this scheme can only be applied to encrypted files and used in a limited way. When the keyed hashes are used up, this scheme will give the user additional online burden. Worku et al. [9] redefined a new integrity protection model that ensured a stronger definition of integrity by adding a second query phase and also proposed a scheme that claims to be able to acquire provable security under the model. However, Liu et al. [11] proved that the scheme in [9] does not satisfy the definition of its own security mode. Worku's scheme cite 10 selects a unique identifier *name* for each file, but the *name* cannot be well embedded into the scheme and is not tightly connected to the scheme. Therefore, the adversary can extract important knowledge in the second inquiry phase of the security model [18].

2.2. Public Audition. Wang et al. [6] proposed a public audition scheme with the privacy protection property that an adversary cannot obtain the information of data in the PDP model. Worku et al. [9] presented that the Wang's scheme [6] is not secure against the attacks from malicious servers and proposed a privacy preserving scheme. Wang et al. [8] proposed an improved scheme to achieve the property of privacy protection. However, these schemes [6, 8, 9] cannot satisfy the definition of the strict privacy protection model, IND-privacy (indistinguishability-privacy), presented by Fan et al. [10]. The IND-privacy model achieves privacy protection by proving the indistinguishability of responses in the auditing process to external auditors. The general idea and design of Worku's scheme [9] and Wang's scheme [8] are similar. The former uses $\mu = \mu' + rh(R)$ to hide the linear combination μ' of the blocks, while the latter is implemented by $\mu = r + h(R)\mu'$, where r and R are the random numbers and h stands for hash function. These two ways can only ensure that external auditors cannot obtain the relevant information of the data, but for the malicious auditors, it is easy to distinguish the different information to obtain the relevant knowledge of the cloud users. The protocol proposed by Fan et al. [10] is inefficient and its symmetric external Diffie-Hellman assumption can be solved in the presence of bilinear mapping.

Table 1: Schemes attribute comparison.

	Ateniese-[3]	Juels-[4]	[5]	Wang-[6]	Wang-[8]	Worku-[9]	Our scheme
Data integrity protection	Yes	Yes	Yes	No	Yes	No	Yes
Public auditability	Yes	No	Yes	No	Yes	Yes	Yes
Privacy protection	No	No	No	No	Yes	Yes	Yes
Blockless verification	Yes	No	Yes	Yes	Yes	Yes	Yes
Batch auditing	No	No	No	No	Yes	Yes	Yes
Retrievability	No	Yes	No	No	No	No	Yes
Audit times	Unlimit	limit	limit	Unlimit	Unlimit	Unlimit	Unlimit
Communication	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$

2.3. Dynamic Maintenance. In practical settings, supporting dynamic maintenance is desired in remote data attestation. Ateniese et al. [19] proposed the concept of dynamic operation and built a scheme that does not require batch encryption based on a symmetric cryptosystem. However, this scheme is limited to the number of queries and does not support full sense of dynamic scenarios. Wang et al. [7, 8] supported full dynamic operations by using Merkle hash tree (MHT) structure. Erway et al. [20] developed a skiplist-based scheme to enable the integrity of data with full dynamics operations. Sookhak et al. [21] also proposed a dynamic scheme. Xin et al. [22] proposed an effective and secure access control approach for multiauthority cloud storage systems.

In addition, the communication and computational overheads are also critical metrics to evaluate the efficiency of cloud services [23, 24]. Among the schemes above, the communication overhead of [8] during validation is $O(n)$, while [9] is more efficient in computational overhead. We illustrate the comparison of the relevant solutions in Table 1.

3. Preliminaries

3.1. Bilinear Map and GDH Groups. Let G_1, G_2, and G_T be multiplicative cyclic groups of the same large prime order p, where $|G_1| = |G_2| = |G_T|$. Let g_1, g_2 be the generators of G_1, G_2, respectively. $e : G_1 \times G_2 \longrightarrow G_T$ is a bilinear map if it satisfies the following properties:

(i) Bilinear: $\forall u \in G_1, v \in G_2$ and $\forall a, b \in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$ holds.

(ii) Nondegenerate: $\exists u \in G_1, v \in G_2$, such that $e(u, v) \neq I_{G_T}$, I_{G_T} is the identity element of the cyclic group G_T.

If there exists an isomorphism $\psi : G_2 \longrightarrow G_1$, with $\psi(g_2) = g_1$, (G_1, G_2) is a Gap Diffie-Hellman (GDH) group pair. We can set $G_1 = G_2 = G$ and $g_1 = g_2 = g$ and take ψ to be the identity map.

(a) Computational co-Diffie-Hellman (co-CDH) on GDH [16]. Given $g_2, g_2^\alpha \in G_2$ and $h \in G_1$ as input and $\alpha \in Z_p$, compute $h^\alpha \in G_1$.

(b) Decision co-Diffie-Hellman (co-DDH) on GDH [16]. Given $g_2, g_2^\alpha \in G_2$ and $h, h^\beta \in G_1$ as input, if $\alpha = \beta$ the output is `yes`; otherwise, the output is `no`. When the answer is `yes`, we say that $(g_2, g_2^\alpha, h, h^\alpha)$ is a co-Diffie-Hellman tuple.

When $G_1 = G_2$, these problems reduce to a standard CDH and DDH problems. The co-DDH problem is easy to be solved but co-CDH is hard on the GDH group [16].

3.2. BLS Signature. The BLS signature [16] includes three functions, **KeyGen**, **Sign**, and **Verify**. Let (G_1, G_2) be a GDH group where $|G_1| = |G_2| = p$. It makes use of a full-domain hash function $H : \{0,1\}^* \longrightarrow G_1$.

(i) **KeyGen.** Randomly choose $x \longleftarrow Z_p$ and compute $v \longleftarrow g_2^x$. The public key is $v \in G_2$ and the private key is x.

(ii) **Sign.** Given a private key $x \in Z_p$ and a message $M \in \{0,1\}^*$, compute $h \longleftarrow H(M) \in G_1$ and $\sigma \longleftarrow h^x$. The signature is $\sigma \in G_1$.

(iii) **Verify.** Given a public key $v \in G_2$, a message $M \in \{0,1\}^*$, and the signature $\sigma \in G_1$, compute $h \longleftarrow H(M) \in G_1$ and verify whether (g_2, v, h, σ) is a valid co-Diffie-Hellman tuple. If so, output *valid*; otherwise, output *invalid*.

4. Approach Overview

4.1. System Model. Our scheme, EoCo, is built on the system model presented in Figure 1. The model consists of three entities, *Users*, *CSP Servers*, and *The Third Party Auditor*.

(i) Users. Cloud users own the data and want to save local storage and computing resources by uploading them to the cloud.

(ii) CSP Servers. The CSP servers have a large amount of storage space available for users. At the same time, CSP servers provide effective cloud operations such as data update and queries and retrieve requests from the customers.

(iii) The Third Party Auditor (TPA). The TPA has more ability and expertise than the clients. Users could ask a TPA help to audit the integrity of the outsourced data on behalf of them.

In our system model, our remote data integrity protection scheme consists of five critical operations, *KeyGen*, *TokenGen*, *Challenge*, *Response*, and *CheckProof*. A cloud user runs **KeyGen** to generate her/his public key pk and private key sk (with a security parameter k as its input). The user

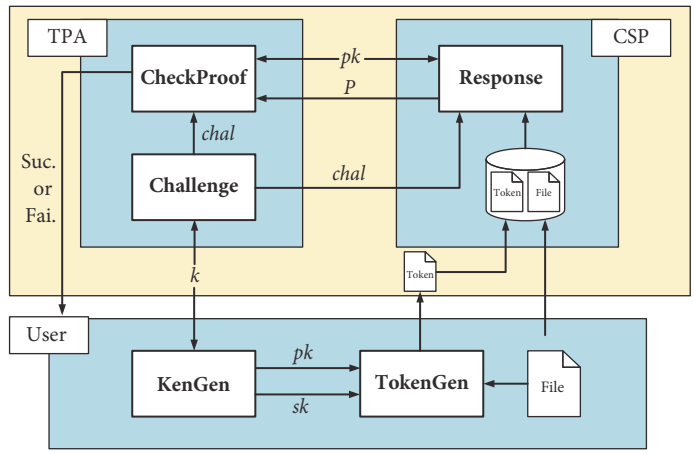


Figure 1: The EoCo-based network system model.

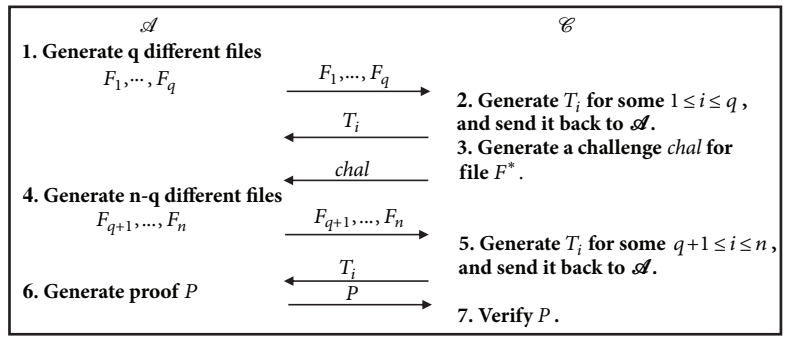


Figure 2: Integrity protection game.

runs the **TokenGen** algorithm to generate a token for the file and outsources the file with corresponding token to the CSP server. When the user wants to check the correctness of her/his data, she/he may delegate the integrity audition to a TPA. The TPA creates *chal* using **Challenge** and sends it to the CSP server. The server runs **Response** and returns the proof P to TPA. The TPA runs **CheckProof** to check whether it is correct, and if it is correct, the output is *success*; otherwise, the output is *failure*.

In the system shown in Figure 1, we mainly take two threats in to account, *the threat to integrity,* and *the threat to privacy* [25]. Accordingly, we classify attackers into two types based on the knowledge H that they processed.

(i) **Threat to Integrity.** The attacker observes the data DA, the authentication identifier T, and the public key pk, i.e., $H =< DA, T, pk >$. The purpose of such attacker is to produce a legitimate proof for the forge DA.

(ii) **Threat to Privacy.** The attacker observes only the public key pk and the proof P, i.e., $H =< P, pk >$. The purpose of such attacker is to acquire additional knowledge, such as the content of data or the type of data.

4.2. Integrity Protection Model. We improve the integrity protection model proposed by Worku et al. [9] and through this model, we demonstrate two objectives.

(1) If the data are not stay the original state, an adversary cannot successfully construct a valid proof in polynomial time with nonnegligible probability.

(2) If the adversary can always pass the verification, then it can be shown that the data remains intact.

Our integrity protection model allows an adversary to query large files F_i. The adversary $\mathscr{A}$, may be a dishonest cloud service provider who interacts with a challenger $\mathscr{C}$ (users or TPA). The integrity game consists of the following steps and we illustrate the details in Figure 2.

(i) **Setup.** The challenger $\mathscr{C}$ runs the algorithm of key generation, sends the public key pk to the adversary $\mathscr{A}$, and retains the private key sk secret.

(ii) **Query1.** The adversary $\mathscr{A}$ adaptively makes tagging queries: it selects a file F_1 and sends it to $\mathscr{C}$. The challenger $\mathscr{C}$ then computes the token T_1 and sends it back to $\mathscr{A}$. The adversary $\mathscr{A}$ continues to query $\mathscr{C}$ for the token $T_2, \ldots, T_q$ on the files of its choice $F_2, \ldots, F_q$. In general, the challenger $\mathscr{C}$ generates T_i for some $1 \le i \le q$.

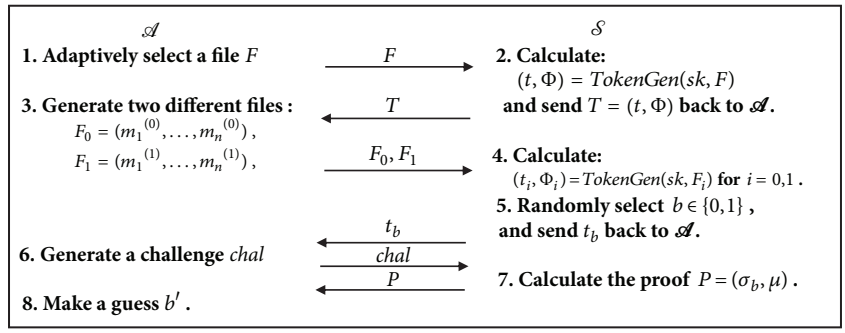


Figure 3: ZK-privacy game.

(iii) **Challenge.** The challenger $\mathscr{C}$ generates a challenge *chal* and requests the adversary $\mathscr{A}$ to provide a proof of integrity for the file F^* (note that file F^* must differ from the F_i in query phase).

(iv) **Query2.** Repeat Query1, the challenger $\mathscr{C}$ generates T_i for some i, $q + 1 \leq i \leq n$.

(v) **Forge.** The adversary $\mathscr{A}$ computes a proof of integrity P for the file F^* according to challenge *chal* and returns P to $\mathscr{C}$.

If the P can pass the verification, the adversary $\mathscr{A}$ wins the game.

Definition 1. An EoCo scheme guarantees data integrity if for any polynomial time adversary $\mathscr{A}$ cannot win the game with nonnegligible probability.

4.3. Privacy Protection Model. In this subsection, we define a new privacy protection model, *the ZK-privacy model*, and prove that the scheme does not have any information leakage by showing that the attacker has zero knowledge during the audit process. The ZK-privacy model takes place between a challenger $\mathscr{S}$ such as cloud server and an adversary $\mathscr{A}$ such as malicious TPA. The game model includes the following steps and we summarize the critical operations in Figure 3.

(i) **Setup.** The challenger $\mathscr{S}$ runs the algorithm of key generation to generate key pair (pk, sk) and sends the public key pk to the adversary $\mathscr{A}$.

(ii) **Phase 1 (Steps 1–3).** The adversary $\mathscr{A}$ adaptively makes queries: it selects a file F and sends it to the challenger $\mathscr{S}$, then the challenger $\mathscr{S}$ generates the token, $T = (t, \Phi)$, for the file F, and sends it to the adversary $\mathscr{A}$.

(iii) **Phase 2 (Steps 4–7).** The adversary $\mathscr{A}$ chooses two files F_0, F_1, and $F_0 \neq F_1$, which are different from the files in Phase 1. Then the challenger $\mathscr{S}$ generates corresponding $T_0 = (t_0, \Phi_0), T_1 = (t_0, \Phi_0)$. Next, the challenger $\mathscr{S}$ randomly selects $b \in \{0, 1\}$ and sends t_b to $\mathscr{A}$. The adversary $\mathscr{A}$ generates a challenge *chal* to $\mathscr{S}$. The challenger $\mathscr{S}$ generates proof, $P_b = (\sigma_b, \mu)$, to the $\mathscr{A}$. Finally, the adversary $\mathscr{A}$ outputs a bit b' as a guess of the b. If $b' = b$, the adversary wins the game.

Definition 2. We define the advantage of the adversary $\mathscr{A}$ as

$$Adv_{\mathscr{A}}(\lambda) = \left| View_{\mathscr{A}}(b = 0) - View_{\mathscr{A}}(b = 1) \right| \quad (1)$$

The $View_{\mathscr{A}}(b = 0 \quad or \quad 1)$ indicates the probability distribution of the view of $\mathscr{A}$ in the case of $b = 0$ or $b = 1$. An EoCo scheme guarantees that there is no an information leakage if $Adv_{\mathscr{A}}(\lambda) = 0$.

5. Our Schemes

5.1. Definition and Framework. Our scheme mainly consists of five polynomial time algorithms: KeyGen, TokenGen, Challenge, Response, and CheckProof. The cloud user is represented by C, the cloud server is represented by S, and the third party auditor is represented by TPA.

(i) $KeyGen(1^k) \longrightarrow (pk, sk)$. *KeyGen* is a probabilistic key generation algorithm, which is set up and initialized by C. It takes as input a security parameter k and outputs a pair of key (pk, sk).

(ii) $TokenGen(sk, pk, F) \longrightarrow T$. *TokenGen* is a deterministic algorithm to compute tokens. It takes as inputs a private key sk, a public key pk and a file $F = (m_1, \ldots, m_n)$. The output is the token $T = (t, \Phi = \{\sigma_i\}_{1 \leq i \leq n})$, where t is a file tag which includes a file *name*, Φ is an ordered collection, and σ_i is the unique authentication identifier corresponding to the blocks m_i in the file F.

(iii) $Challenge(1^k) \longrightarrow chal$. *Challenge* is a deterministic algorithm. It takes as input a security parameter k. It outputs a challenge *chal*.

(iv) $Response(pk, chal, \Phi, F) \longrightarrow P$. *Response* is to create the proof of the data integrity corresponding to the received challenge. It takes as inputs a public key pk, a file F, a challenge *chal*, and an ordered set Φ. The output is an integrity proof P of the file F.

(v) $CheckProof(pk, P, chal, t) \longrightarrow (suc.\ or\ fail.)$. *CheckProof* is the function to verify the returned proof P. It takes as inputs a public key pk, a challenge *chal*, and a proof P as input. It outputs *success* or *failure*.

Table 2: Notation list of symbols and parameters.

Symbols	Description
p	a large prime
Z_p	a group of integers with order p
G	a cyclic group
g	an element of G
H	a hash function, $\{0,1\}^* \longrightarrow G$
π	a pseudo-random permutation, $\{0,1\}^{\log_2(n)} \times K \longrightarrow \{0,1\}^{\log_2(n)}$
f	a pseudo-random functions, $\{0,1\}^* \times K \longrightarrow Z_p$
$SSig(\cdot)$	an additional BLS signature
$F = (m_1, ..., m_n)$	divide a big data file F into n small file blocks and $m_i \in Z_p$

A EoCo scheme can be summed up as two processes: one is the setup process to initialize the scheme, and the other is the verification process to confirm whether the data is integrity.

(i) **Setup.** The cloud user C has a file F and runs the algorithm **KeyGen** to generate the key pair (pk, sk). C then stores the key pair locally and uses it to run the algorithm **TokenGen** to generate a token, $T = (t, \Phi = \{\sigma_i\}_{1 \le i \le n})$. Finally, C sends T and F to S and deletes F and T from its local storage.

(ii) **Audition.** TPA runs the algorithm *Challenge* to generate a challenge *chal* corresponding to the target files. TPA sends *chal* to the server S. After receiving *chal*, S computes the proof P accordingly and returns the TPA the proof P and t by using the function *Response*. The TPA validates P and t using the function *CheckProof*.

5.2. Notations. In this subsection, we illustrate the parameters and notations used in our scheme. As listed in Table 2, p is a large prime and the EoCo scheme is built on the group Z_p and supports a cyclic group G with a bilinear setting. Let g be an element of cyclic group. In the setup phase, the date owner splits the file F into n small file blocks, that is, $F = (m_1, \ldots, m_n)$, and for $1 \le i \le n$, every block $m_i \in Z_p$. $H : \{0,1\}^* \longrightarrow G$ is a secure hash function. The additional BLS signature used in our scheme is represented as $SSig(\cdot)$. In addition, we take advantage of a pseudorandom permutation (PRP) π and a pseudorandom functions (PRF) f with the following parameters. We write $f_k(x)$ to denote f keyed with key k applied on input x.

(i) $\pi : \{0,1\}^{\log_2(n)} \times k \longrightarrow \{0,1\}^{\log_2(n)}$.

(ii) $f : \{0,1\}^* \times k \longrightarrow Z_p$.

5.3. EoCo Scheme

(i) **KeyGen**(1^k)**.** The algorithm first generates a secure BLS signature key pair (spk, ssk). Next, the algorithm randomly selects an element in the Z_p, $x \longleftarrow Z_p$, and computes $v \longleftarrow g^x \in G$. Then, the algorithm randomly selects an element in the G, $u \longleftarrow G$, and computes $w \longleftarrow u^x$. Finally, the algorithm returns public key $pk = (spk, g, v, u, w)$ and private key $sk = (ssk, x)$.

(ii) **TokenGen**$(sk, pk, F = (m_1, \ldots, m_n))$**.** The algorithm randomly selects an element in the Z_p as a unique identifier of file F, $name \longleftarrow Z_p$, and calculates file identification tag by BLS signature, $t = name \parallel SSig_{ssk}(name)$. Next, the algorithm generates unique authentication identifier for each block m_i in the file, $\sigma_i \longleftarrow (H(W_i) \cdot u^{m_i})^x \in G$. For $1 \le i \le n$, $W_i = name \parallel i$. The algorithm then outputs token $T = (t, \Phi = \{\sigma_i\}_{1 \le i \le n})$.

(iii) **Challenge**(1^k)**.** The algorithm first determines the number of file blocks c. According to the security parameters k, the algorithm randomly selects key k_1 for π and key k_2 for f. Finally, the algorithm returns $chal = (c, k_1, k_2)$.

(iv) **Response**$(pk, chal, \Phi = \{\sigma_i\}_{1 \le i \le n}), F)$**.** The algorithm first selects a secret random value $r \longleftarrow Z_p$. For $1 \le j \le c$, the algorithm computes $i_j = \pi_{k_1}(j)$ and $a_j = f_{k_2}(j)$, and the algorithm then computes aggregated authentication identifier $\sigma = \sigma_{i_1}^{a_1} \cdot \sigma_{i_2}^{a_2}, \ldots, \cdot\sigma_{i_c}^{a_c} \cdot w^r$ (note that σ_{i_j} is the i_j-th value in Φ). $\mu = v^{\mu' + r} \in G$, and $\mu' = a_1 m_{i_1} +, \ldots, a_c m_{i_c}$. Finally, the algorithm outputs proof $P = (\sigma, \mu)$.

(v) **CheckProof**$(pk, P, chal, t)$**.** The algorithm first verifies whether the file tag t is correct. If t is incorrect, the algorithm outputs *failure* and quit; otherwise the algorithm extracts *name* and for $1 \le j \le c$, the algorithm calculates $i_j = \pi_{k_1}(j)$, $a_j = f_{k_2}(j)$ and $W_{i_j} = name \parallel i_j$. Finally, the algorithm verifies the following equation:

$$e(\sigma, g) = e\left(\prod_{j=1}^{c} H\left(W_{i_j}\right)^{a_j}, v\right) \cdot e(u, \mu) \tag{2}$$

If (2) holds, *CheckProof* returns *success*; otherwise, it returns *failure*.

6. Scheme Analysis

6.1. Correctness. In this subsection, we discuss the correctness of our scheme; that is, the CSPs will definitely pass through the audition if they follow the protocol honestly. By signing the *name* of a file with an additional BLS signature,

we can prevent an adversary from tampering with the *name*. We use a hash function to ensure that the file tag, *name*, is perfectly embedded in the authentication identifier. We make use of a PRP function and a PRF function to ensure the randomness of the challenge content and the security of the response. If the data is kept properly, the correctness of the verification equation (2) can be shown as follows:

$$\begin{aligned} & e\left(\prod_{j=1}^{c} H\left(W_{i_j}\right)^{a_j}, v\right) \cdot e(u, \mu) \\ &= e\left(\prod_{j=1}^{c} H\left(W_{i_j}\right)^{a_j}, g^x\right) \cdot e\left(u^{\mu'+r}, g^x\right) \\ &= e\left(\prod_{j=1}^{c} H\left(W_{i_j}\right)^{a_j} u^{\mu'}, g^x\right) \cdot e\left(u^r, g^x\right) \\ &= e\left(\prod_{j=1}^{c} \sigma_{i_j}^{a_j}, g\right) \cdot e\left(\left(u^x\right)^r, g\right) = e\left(\prod_{j=1}^{c} \sigma_{i_j}^{a_j} w^r, g\right) \\ &= e(\sigma, g) \end{aligned} \tag{3}$$

6.2. Integrity Protection. The integrity protection of EoCo is based on the symmetric co-CDH problem or standard CDH problem.

Theorem 3. *Under the CDH assumptions, EoCo guarantees data integrity protection in the random oracle model.*

*We first propose a simplified scheme, S-EoCo. By proving the security of S-EoCo, we can prove the security of EoCo. S-EoCo and EoCo differ only in the **KenGen** and **Response** algorithms as follows:*

(i) ***KeyGen***(1^k)***.*** *The algorithm first generates a secure BLS signature key pair* (spk, ssk)*. Next, the algorithm randomly selects an element in the* Z_p*,* $x \longleftarrow Z_p$*, and computes* $v \longleftarrow g^x \in G$*. Then, the algorithm randomly selects an element in the G,* $u \longleftarrow G$*. Finally, the algorithm returns public key* $pk = (spk, g, v, u)$ *and private key* $sk = (ssk, x)$*.*

(ii) ***Response***$(pk, chal, \Phi = \{\sigma_i\}_{1 \le i \le n}), F)$***.*** *For* $1 \le j \le c$*, the algorithm computes* $i_j = \pi_{k_1}(j), a_j = f_{k_2}(j)$*. Then, for* $I = \{i_1, i_2, \ldots, i_c\}$*, the algorithm computes aggregated authentication identifier* $\sigma = \sigma_{i_1}^{a_1} \cdot \sigma_{i_2}^{a_2}, \ldots, \cdot \sigma_{i_c}^{a_c}$*,* $\mu = v^{\mu'} \in G$*, where* $\mu' = a_1 m_{i_1} +, \ldots, a_c m_{i_c}$*. Finally, the algorithm outputs proof* $P = (\sigma, \mu)$*.*

Proof. We reduce the security of our S-EoCo scheme to the security of the CDH problem. We model hash function $H(\cdot)$ as random oracle.

If an adversary $\mathscr{A}$ can break the integrity protection of the S-EoCo scheme, we show how to construct an adversary $\mathscr{C}$ that uses $\mathscr{A}$ in order to break CDH problem.

For the CDH problem, $\mathscr{C}$ is given $(g, g^\alpha, h) \in G$ and needs to calculate h^α. Then $\mathscr{C}$ will play the role of the challenger in the integrity protection game and will interact with $\mathscr{A}$ as follows.

(i) **Setup.** The challenger $\mathscr{C}$ selects a secret key pair (spk, ssk) and gets $pk = (spk, g, v, u)$, where $u = g^a h^b$, $a \longleftarrow Z, b \longleftarrow Z_p$, $v = g^\alpha$. Then $\mathscr{C}$ sends public key to the adversary $\mathscr{A}$.

(ii) **Query1.** The adversary $\mathscr{A}$ adaptively presents queries in the way that it selects different files $F_1, \ldots, F_n$ and sends them to $\mathscr{C}$ to create *Token*s. In order to answer queries, $\mathscr{C}$ simulates in a random oracle machine as follows:

(1) For $F = \{m_1, m_2, \ldots, m_n\}$, randomly selects an element in group Z_p as the unique identifier of file F, $name \longleftarrow Z_p$, and calculates file identification tags $t = name \parallel SSig_{ssk}(name)$.

(2)

(a) When $\mathscr{A}$ queries j for hash value, $\mathscr{C}$ first checks whether j is in the hash tuple list $(j, H(j))$. If it is in the list, $\mathscr{C}$ sends $H(j)$ to $\mathscr{A}$ as the answer; otherwise, $\mathscr{C}$ randomly selects and replies $\mathscr{C}$ an elements in G, $\omega \longleftarrow G$, as $H(j)$, and adds $H(j)$ to the hash tuple list $(j, H(j))$.

(b) When $\mathscr{A}$ performs the identifier query, $\mathscr{C}$ calculates $W_i = name \parallel i$. If $W_i = j$, $\mathscr{C}$ announces *failure*; otherwise, $\mathscr{C}$ randomly selects $r_i \longleftarrow Z_p$, calculates $H(W_i) = g^{r_i} \cdot g^{-am_i} \cdot h^{-bm_i}$, and builds a list of identifiers $(W_i, m_i, r_i, H(W_i), \sigma_i)$. $\mathscr{C}$ can compute σ_{m_i} using the formula below.

$$\begin{aligned} \sigma_{m_i} &= \left(H(W_i) \cdot u^{m_i}\right)^\alpha = \left(g^{r_i} \cdot g^{-am_i} \cdot h^{-bm_i} \cdot u^{m_i}\right)^\alpha \\ &= \left(g^{r_i} \cdot g^{-am_i} \cdot h^{-bm_i} \cdot g^{am_i} \cdot h^{bm_i}\right)^\alpha = \left(g^{r_i}\right)^\alpha \\ &= \left(g^\alpha\right)^{r_i} \end{aligned} \tag{4}$$

(3) $\mathscr{C}$ sends the file identifier $T = (t, \Phi = \{\sigma_i\}_{1 \le i \le n})$ to $\mathscr{A}$.

(iii) **Challenge.** For files $F^* = \{m_1^*, m_2^*, \ldots, m_n^*\}$, $\mathscr{C}$ sends challenge *chal* to $\mathscr{A}$, and F^* is different from above F that have been query.

(iv) **Query2.** Repeat the Query1, but $\mathscr{A}$ cannot query the file blocks included in *chal*.

(v) **Forge.** $\mathscr{A}$ creates and returns P to $\mathscr{C}$ according to challenge *chal*. If P can pass through the verification, $\mathscr{A}$ wins the game.

Suppose that an honest cloud server computes a proof, $P = (\sigma, \mu)$, then the following verification formula should be satisfied:

$$e(\sigma, g) = e\left(\prod_{i \in I} H(W_i)^{a_i}, v\right) \cdot e(u, \mu) \tag{5}$$

Suppose that the adversary$\mathscr{A}$ wins games with nonnegligible probability. The proof output by $\mathscr{A}$ is $P^* = (\sigma^*, \mu^*)$ that

can pass the verification with nonnegligible probability, thus formula (6) is satisfied.

$$e(\sigma^*, g) = e\left(\prod_{i\in I} H(W_i)^{a_i}, v\right) \cdot e(u, \mu^*) \tag{6}$$

According to the game, $\mu \neq \mu^*$. Let $\Delta\mu' = \mu'^* - \mu'$. Because G is a multiplicative cyclic group, we can find an inverse element of σ in the group. We substitute σ^{-1} for σ in formula (5) to get the deformation of formula (5).

$$e\left(\sigma^{-1}, g\right) = e\left(\prod_{i\in I} H(W_i)^{a_i}, v\right)^{-1} \cdot e\left(u, \mu^{-1}\right) \tag{7}$$

We multiply σ^* by σ^{-1} and obtain the following equation:

$$\begin{aligned} e\left(\sigma^*\sigma^{-1}, g\right) &= e\left(u, \mu^*\mu^{-1}\right) = e\left(u, v^{\mu'^* - \mu'}\right) \\ &= e\left(u, v^{\Delta\mu'}\right) = e\left(u^{\Delta\mu'}, v\right) \end{aligned} \tag{8}$$

Because $u = g^a h^b$, we replace u in formula (8).

$$\begin{aligned} e\left(\sigma^*\sigma^{-1}, g\right) &= e\left(\left(g^a h^b\right)^{\alpha\Delta\mu'}, g\right) \\ &= e\left(\left(g^a\right)^{\alpha\Delta\mu'}\left(h^b\right)^{\alpha\Delta\mu'}, g\right) \\ &= e\left(v^{a\Delta\mu'}\left(h^\alpha\right)^{b\Delta\mu'}, g\right) \end{aligned} \tag{9}$$

The above equation can be further deformed as follows:

$$e\left(\sigma^*\sigma^{-1}v^{-a\Delta\mu'}, g\right)^{-b\Delta\mu'} = e\left(h^\alpha, g\right) \tag{10}$$

It can be derived from formula (10) that $h^\alpha = (\sigma^*\sigma^{-1}v^{-a\Delta\mu'})^{-b\Delta\mu'}$. $\mathscr{C}$ takes back μ'^* from the adversary $\mathscr{A}$ and calculates $\Delta\mu' = \mu'^* - \mu'$. It means that as long as $b\Delta\mu' \neq 0 \mod p$ $\mathscr{C}$ can solve the CDH problem. As $\Delta\mu' \neq 0 \mod p$, the probability that $b\Delta\mu' \neq 0$ is $1 - 1/p$, which is nonnegligible. And the probability that simulation failure possibility of $\mathscr{C}$ is $1/q$ is negligible. So if the adversary wins the game with a nonnegligible probability, the CDH problem can be solved with a nonnegligible probability. Aggregation tags in **Response** in the EoCo scheme is that $\sigma = \sigma_{i_1}^{a_1} \cdot \sigma_{i_2}^{a_2}, \ldots, \cdot\sigma_{i_c}^{a_c} \cdot w^r$ and $\mu = v^{\mu'+r} \in G$. We add w^r on the basis of S-EoCo that makes scheme stronger.

Here we proved the Theorem 3. □

6.3. Privacy Preserving

Theorem 4. *According to Definition 2, an adversary is completely zero knowledge if, for any polynomial time algorithm $\mathscr{A}$, $Adv_{\mathscr{A}}(\lambda) = 0$.*

Proof. To prove Theorem 4, we construct a simulator $\mathscr{S}$ to interact with an adversary $\mathscr{A}$ following the steps below.

(1) $\mathscr{A}$ selects two different files $F_0 = (m_1^{(0)}, \ldots, m_n^{(0)})$ and $F_1 = (m_1^{(1)}, \ldots, m_n^{(1)})$, for $1 \leq i \leq n$, which satisfy $m_i^{(0)} \neq m_i^{(1)}$.

(2) The simulator $\mathscr{S}$ generates $T_0 = (t_0, \Phi_0)$ and $T_1 = (t_1, \Phi_1)$ for the files F_0 and F_1, respectively. Then $\mathscr{S}$ randomly selects $b \in \{0, 1\}$ and sends t_b to $\mathscr{A}$.

(3) When $\mathscr{A}$ received t_b, $\mathscr{A}$ randomly generates a challenge $chal = \{c, k_1, k_2\}$ and sends $chal$ to $\mathscr{S}$ request proof of F_b

(4) The simulator $\mathscr{S}$ calculates and sends response proof $P_b = (\sigma_b, \mu)$ to $\mathscr{A}$.

(5) The identifier in P_b is that $\sigma_b = \sigma_{i_1}^{a_1} \cdot \sigma_{i_2}^{a_2}, \ldots, \cdot\sigma_{i_c}^{a_c} \cdot w^r$, $\mu = v^{\mu'+r} \in G$, and $\mu' = a_1 m_{i_1}^{(b)} + \ldots, a_c m_{i_c}^{(b)}$, and it satisfies

$$e(\sigma, g) = e\left(\prod_{i\in I} H(W_i)^{a_i}, v\right) \cdot e(u, \mu) \tag{11}$$

From the perspective of $\mathscr{A}$, $\mathscr{A}$ can only see (P_b, F_0, F_1, pk). Since $\mathscr{A}$ cannot know the value of r (unless $\mathscr{A}$ can solve the discrete logarithm (DL) problem) and r is randomly and uniformly distributed, for the adversary $\mathscr{A}$, P_0 and P_1 also are uniformly distributed.

Since pk is public, so $View_{\mathscr{A}}(P_0, F_0, F_1, pk)$ and $View_{\mathscr{A}}(P_1, F_0, F_1, pk)$ have the same probability distribution. The advantage of the adversary $\mathscr{A}$ is as follows:

$$\begin{aligned} Adv_{\mathscr{A}}(\lambda) = \big|&Pr\left[View_A\left(P_0, F_0, F_1, pk\right)\right] \\ &- Pr\left[View_A\left(P_1, F_0, F_1, pk\right)\right]\big| = 0 \end{aligned} \tag{12}$$

The above equation shows that the protocol is strict zero knowledge to the adversary $\mathscr{A}$, and thus, the adversary cannot get any information through the auditing process.

Here we complete the proof. □

6.4. Batch Auditing. Sometimes, a TPA will audit multiple files on behalf of different users. While one by one validation is too tedious and inefficient, it is desired that the TPA could parallelly audit the integrity of multiple cloud user data. Suppose that Q cloud users are entrusted with the same TPA. We slightly modify our scheme by using BLS signature to aggregate different signatures and thus provide the effective verification for all Q users/files simultaneously. In fact, our attribute of batch auditing is to enhance the audit efficiency of TPA. Therefore, in EoCo scheme, we only need to make minor modifications on the **CheckProof** algorithm.

(i) **CheckProof**$(pk, P, chal, t)$**.** For $1 \leq l \leq Q$, the algorithm first verifies whether the file tag t_l is correct. The algorithm then outputs *failure* if it is wrong; otherwise, the algorithm extracts $name_l$ and continues. Next, for $1 \leq j \leq c$, the algorithm calculates $i_j = \pi_{k_1}(j), a_j = f_{k_2}(j)$ and $(W_{i_j})_l = name_l \parallel i_j$ and computes aggregate authentication

Table 3: Communication complexity comparison.

	SW-[5]	Wang-[8]	Worku-[9]	Our scheme
Challenge form and the number of bits required	$\{(i, v_i)\}_{1\le i\le n}$	$\{(i, v_i)\}_{1\le i\le n}$	(c, k_1, k_2)	(c, k_1, k_2)
	$2n\|p\|$bits	$2n\|p\|$bits	$3\|p\|$bits	$3\|p\|$bits
Response form and the number of bits required	(σ, μ)	(σ, μ, R)	(σ, μ, R)	(σ, μ)
	$2\|p\|$bits	$3\|p\|$bits	$3\|p\|$bits	$2\|p\|$bits
Entire Overhead	$O(n)$	$O(n)$	$O(1)$	$O(1)$

identifier $\sigma_Q = \prod_{l=1}^{Q} \sigma_l$. Finally, the algorithm verifies the following equation:

$$\prod_{l=1}^{Q} \left\{ e\left(\prod_{j=1}^{c} H\left(\left(W_{i_j} \right)_l \right)^{a_j}, v_l \right) \cdot e\left(u, \mu_l\right) \right\} = e\left(\sigma_Q, g\right) \tag{13}$$

We validate the correctness of batch audition scheme below and show that the honest CSPs will pass the audition successfully if they follow the protocol.

$$\begin{aligned} &\prod_{l=1}^{Q} \left\{ e\left(\prod_{j=1}^{c} H\left(\left(W_{i_j} \right)_l \right)^{a_j}, v_l \right) \cdot e\left(u, \mu_l\right) \right\} \\ &= \prod_{l=1}^{Q} \left\{ e\left(\prod_{j=1}^{c} H\left(W_{i_j} \right)_l^{a_j} u^{\mu_l'}, g^{x_l} \right) \cdot e\left(u^{r_l}, g^{x_l}\right) \right\} \\ &= \prod_{l=1}^{Q} \left\{ e\left(\prod_{j=1}^{c} \left(\sigma_{i_j} \right)_l^{a_j}, g \right) \cdot e\left(\left(u^{x_l}\right)^{r_l}, g \right) \right\} \\ &= \prod_{l=1}^{Q} \left\{ e\left(\prod_{j=1}^{c} \left(\sigma_{i_j} \right)_l^{a_j}, g \right) \cdot e\left(\left(w_l\right)^{r_l}, g \right) \right\} \\ &= \prod_{l=1}^{Q} e\left(\prod_{j=1}^{c} \left(\sigma_{i_j} \right)_l^{a_j} \cdot \left(w_l\right)^{r_l}, g \right) = e\left(\prod_{l=1}^{Q} \sigma_l, g \right) \\ &= e\left(\sigma_Q, g\right) \end{aligned} \tag{14}$$

7. Evaluation

In this section, we evaluate the performance of our proposed scheme in the following three aspects:

(1) Computational complexity: the cost of the setup phase, the cost of producing response proof by S, and the cost of verify response proof by an auditor.

(2) Block complexity: S needs the number of file blocks to be accessed according to the challenge.

(3) Communication complexity: the data size and bandwidth of communication between an auditor and a CSP.

Suppose that the EoCo chooses c file blocks for audition and the security parameter is k. The large prime number p should be $2k$ according to the analysis by Shacham et al. [5]. If the security level is 80 bits, then $|p| = 160$ bits. This determines the required block size to achieve the desired security level. We compare the EoCo with the scheme in [5, 8, 9] in terms of computation complexity and communication complexity in this section.

7.1. Communication Complexity. We can see from Table 3 that the number of bits required of SW-[5] is $2n|p| + 2|p|$ bits. The number of bits required of Wang-[8] is $2n|p| + 3|p|$ bits. The number of bits required of Worku-[9] is $6|p|$ bits. The EoCo' the number of bits required is $5|p|$ bits. From this, we can see that the EoCo has the smallest communication overhead, which can greatly reduce the I/O burden of cloud providers and increase bandwidth utilization.

7.2. Computation Complexity. In order to calculate the cost of computing on both sides of the auditor and cloud service provider, we detail the operations on the basic computational symbols in Table 4. We can see comparison of several schemes from Table 5 in three aspects.

(i) *Server computation overhead:* for cloud service providers, we see that the Worku-[9] is obviously less than Wang-[8]. While the EoCo is compared with the Worku-[9], the EoCo adds one more exponential operation, but reduces one hash operation. Generally speaking, one hash is more time-consuming than the exponential operation, so that the EoCo is more efficient on this side.

(ii) *Auditor computation overhead:* for the auditor, Wang-[8] and Worku-[9] are only slightly different in efficiency; the latter uses one-time exponential operation less than the former. Although our scheme uses one-time bilinear mapping more than Worku-[9], our scheme uses one-time hash operation, three-time exponential operation and two-time multiplication operation less than Worku-[9]. In this respect, the EoCo has a great promotion.

(iii) *Setup phase overhead:* because the schemes are based on BLS short signatures, the computational overhead of the setup phase is equal.

According to the theoretical analysis above, our scheme ensures the desired security properties with relatively low performance overhead.

Table 4: Notation of cryptographic operations.

Operations	Description
$Hash^t_{Z_p}$	t map-to-Z_p operations
$Hash^t_G$	t map-to-G operations
$Mult^t_G$	t multiplication in group G
$Mult^t_{Z_p}$	t multiplication in group Z_p
Exp^t_G	t exponentiations in group G
BM^t_G	t bilinear mappings among elements of G
Add^t_G	t additions on group G
$Add^t_{Z_p}$	t additions on group Z_p
$c - MultExp^t_G(l)$	t c-term exponentiations in group G,$l \leq \lvert p \rvert$

Table 5: Computation comparison.

	Our scheme	Wang-[8]	Worku-[9]
Server computation overhead	$Mult^c_G + Exp^2_G +$ $Add^c_{Z_p} + Mult^{c-1}_{Z_p} +$ $c - MultExp^1_G(a_i)$	$Mult^{c-1}_G + BM^1_G +$ $Exp^1_G + Add^c_{Z_p} + Mult^{c+1}_{Z_p} +$ $Hash^1_{Z_p} + c - MultExp^1_G(v_i)$	$Mult^{c-1}_G + Exp^1_G +$ $Add^c_{Z_p} + Mult^{c+1}_{Z_p} +$ $Hash^1_{Z_p} + c - MultExp^1_G(v_i)$
Auditor computation overhead	$Hash^c_G + BM^3_G +$ $c - MultExp^1_G(a_i)$	$Hash^c_G + Mult^2_G + Exp^3_G +$ $BM^2_G + Hash^1_{Z_p} +$ $c - MultExp^1_G(v_i)$	$Hash^c_G + Mult^2_G + Exp^2_G +$ $BM^2_G + Hash^1_{Z_p} +$ $c - MultExp^t_G(l)$
Setup phase overhead	$Mult^n_G + Hash^n_G +$ Exp^{2n}_G	$Mult^n_G + Hash^n_G +$ Exp^{2n}_G	$Mult^n_G + Hash^n_G +$ Exp^{2n}_G

8. Conclusion

In this paper, we proposed a secure and effective cloud data integrity verification scheme with privacy protection, EoCo, which is based on the BLS short signature. We presented a more practical data integrity protection model and introduced a ZK-privacy model to ensure the privacy preserving property in public audition. We theoretically analyzed our approach and prove the security of EoCo based on the CDH problem. We also demonstrated that our scheme ensures zero knowledge leakage through indistinguishable views of the attacker. We evaluated the performance of EoCo and made the comparison with related solutions. The analysis results show that our approach has relatively small communication and computational complexity.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (no. 2015BAG15B01) and the National Natural Science Foundation of China (nos. U17733115, 61402029, 61379002, and 61370190).

References

[1] G. S. Aujla, R. Chaudhary, N. Kumar, A. K. Das, and J. J. P. C. Rodrigues, "SecSVA: Secure Storage, Verification, and Auditing of Big Data in the Cloud Environment," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 78–85, 2018.

[2] L. Xiao, D. Xu, C. Xie, N. B. Mandayam, and H. V. Poor, "Cloud storage defense against advanced persistent threats: a prospect theoretic study," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 3, pp. 534–544, 2017.

[3] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 598–609, Virginia, Va, USA, November 2007.

[4] A. Juels and B. S. Kaliski Jr., "Pors: proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 584–597, ACM, Alexandria, VA, USA, November 2007.

[5] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Advances in Cryptology—ASIACRYPT 2008: Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 2008*, vol. 5350 of *Lecture Notes in Computer Science*, pp. 90–107, Springer, Berlin, Germany, 2008.

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceedings of the IEEE INFO-COM*, pp. 525–533, March 2010.

[7] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public audibility and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel Distributed Systems*, vol. 22, no. 5, pp. 847–859, 2011.

[8] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[9] S. G. Worku, C. Xu, J. Zhao, and X. He, "Secure and efficient privacy-preserving public auditing scheme for cloud storage," *Computers and Electrical Engineering*, vol. 40, no. 5, pp. 1703–1713, 2014.

[10] X. Fan, G. Yang, Y. Mu, and Y. Yu, "On indistinguishability in remote data integrity checking," *The Computer Journal*, vol. 58, no. 4, pp. 823–830, 2013.

[11] H. Liu, L. Chen, Z. Davar, and M. R. Pour, "Insecurity of an efficient privacy-preserving public auditing scheme for cloud data storage," *Journal of Universal Computer Science*, vol. 21, no. 3, pp. 473–482, 2015.

[12] *PASOS. Revista de Turismo y Patrimonio Cultural*.

[13] G. Ateniese, R. Burns, R. Curtmola et al., "Remote data checking using provable data possession," *ACM Transactions on Information and System Security*, vol. 14, no. 1, article 12, 2011.

[14] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: theory and implementation," in *Proceedings of the ACM Workshop on Cloud Computing Security (CCSW '09)*, pp. 43–53, November 2009.

[15] H. Shacham and B. Waters, "Compact proofs of retrievability," *Journal of Cryptology. The Journal of the International Association for Cryptologic Research*, vol. 26, no. 3, pp. 442–483, 2013.

[16] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Advances in Cryptology—ASIACRYPT 2001*, vol. 2248 of *Lecture Notes in Computer Science*, pp. 514–532, Springer, Berlin, Germany, 2001.

[17] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," *Proc. of Hotos07: Workshop on Hot Topics in Operating Systems*, p. 1, 2007.

[18] V. A. Chang, "A model to compare cloud and non-cloud storage of Big Data," *Future Generation Computer Systems*, vol. 57, pp. 56–76, 2016.

[19] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (SecureComm '08)*, pp. 1–10, ACM, Istanbul, Turkey, September 2008.

[20] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS 2009)*, pp. 213–222, ACM, Chicago, Ill, USA, November 2009.

[21] M. Sookhak, A. Gani, M. K. Khan, and R. Buyya, "Dynamic remote data auditing for securing big data storage in cloud computing," *Information Sciences*, 2015.

[22] L. Xin, X. Sun, Z. Fu, L.-A. Zhang, and J. Xi, "Effective and secure access control for multi-authority cloud storage systems," *International Journal of Security and Its Applications*, vol. 10, no. 2, pp. 217–236, 2016.

[23] L. Yu, H. Shen, K. Sapra, L. Ye, and Z. Cai, "CoRE: Cooperative End-to-End Traffic Redundancy Elimination for Reducing Cloud Bandwidth Cost," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 2, pp. 446–461, 2017.

[24] L. Yu, H. Shen, Z. Cai, L. Liu, and C. Pu, "Towards Bandwidth Guarantee for Virtual Clusters under Demand Uncertainty in Multi-Tenant Clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 2, pp. 450–465, 2018.

[25] V. Attasena, J. Darmont, and N. Harbi, "Secret sharing for cloud data security: a survey," *The VLDB Journal*, vol. 26, no. 5, pp. 657–681, 2017.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 8392080, 20 pages
https://doi.org/10.1155/2018/8392080

Research Article

A Compatible OpenFlow Platform for Enabling Security Enhancement in SDN

Haosu Cheng,[1,2] Jianwei Liu,[1,2] Jian Mao ,[1,2] Mengmeng Wang,[1] Jie Chen,[1] and Jingdong Bian[1,2]

[1]*School of Cyber Science and Technology, Beihang University, Beijing, China*
[2]*School of Electronic and Information Engineering, Beihang University, Beijing, China*

Correspondence should be addressed to Jian Mao; maojian@buaa.edu.cn

Received 7 September 2018; Accepted 31 October 2018; Published 15 November 2018

Guest Editor: Chunqiang Hu

Software-defined networking (SDN) is a representative next generation network architecture, which allows network administrators to programmatically initialize, control, change, and manage network behavior dynamically via open interfaces. SDN is widely adopted in systems like 5G mobile networks and cyber-physical systems (CPS). However, SDN brings new security problems, e.g., controller hijacking, black-hole, and unauthorized data modification. Traditional firewall or IDS based solutions cannot fix these challenges. It is also undesirable to develop security mechanisms in such an ad hoc manner, which may cause security conflict during the deployment procedure. In this paper, we propose OSCO (Open Security-enhanced Compatible OpenFlow) platform, a unified, lightweight platform to enhance the security property and facilitate the security configuration and evaluation. The proposed platform supports highly configurable cryptographic algorithm modules, security protocols, flexible hardware extensions, and virtualized SDN networks. We prototyped our platform based on the Raspberry Pi Single Board Computer (SBC) hardware and presented a case study for switch port security enhancement. We systematically evaluated critical security modules, which include 4 hash functions, 8 stream/block ciphers, 4 public-key cryptosystems, and key exchange protocols. The experiment results show that our platform performs those security modules and SDN network functions with relatively low computational (extra 2.5% system overhead when performing AES-256 and SHA-256 functions) and networking performance overheads (73.7 Mb/s TCP and 81.2Mb/s UDP transmission speeds in 100Mb/s network settings).

1. Introduction

Recent years, cyber-physical systems are widely adopted in areas such as personal healthcare, emergency response, traffic flow control, and electric power management. CPS uses emerging technologies such as 5G and Edge computing as its building components. As one of the core technologies in 5G network [1] and Edge computing [2], software-defined-networking (SDN) is applied as a basic networking infrastructure in the cyber-physical system (CPS), which is a collection of interconnected computing devices interacting with the physical world with its users. Being a representative next generation networking technique, SDN decouples the data plane from the control plane and supports programmatically initialization and dynamic network behavior management. Its good performance in flexibility and scalability brings advances in CPS deployment and configuration.

Figure 1 shows the classic SDN architecture which includes three virtual layers, *control plane*, *data plane*, and *application plane* [3, 4]. The application plane invokes software-based logic in the control plane via REST-APIs (Representational State Transfer Application Programming Interface), which is also called *Northbound Interface*. The deployed logic decisions are executed by the data plane through *Southbound Interfaces*, e.g., OpenFlow, a widely adopted implementation architecture of SDN. The network control traffic is transferred from the infrastructure (data plane) to the controller (control plane). With the help of SDN apps, network operators achieve distinguished properties of network control, automation, and resource optimization.

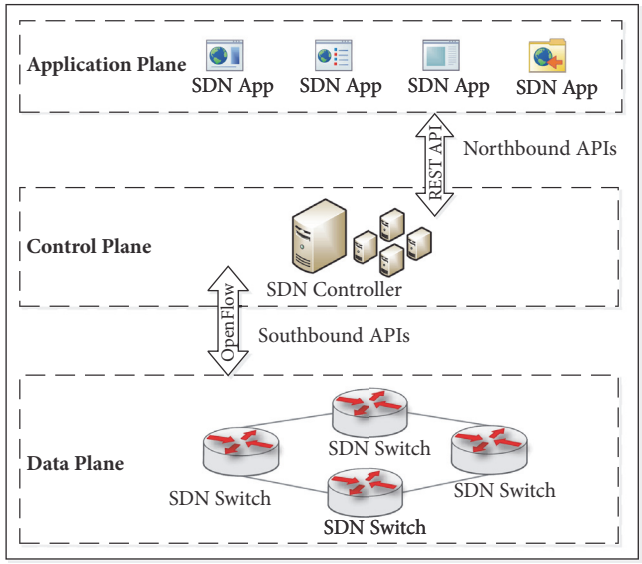


Figure 1: The three planes in SDN architecture.

Table 1: Comparison of SDN attack types.

Attack Type	Targeted SDN Layer
DDOS	Control, Data
DOS	Control, Data
Hijacked Controller	Control, Data, App
Malicious Applications	App
Man-in-the-middle	Control, Data, Control-Data link
Black-hole	Control, Data, Control-Data link
Eavesdropping	Control, Data, App

Although SDN provides a new solution for the network architecture, it exposes many security problems [5]. Since a centralized controller is responsible for managing the entire network, any security compromises of the controller may cause the whole network to crash. Furthermore, security lapses in controller to data plane communication may lead to illegitimate access to network resources. SDN applications enable the users to access network resources, deploy new rules, and manipulate the behavior of the network through the interaction with the control plane. Meanwhile, securing the network from malicious applications or abnormal behavior of applications is a serious security challenge in SDN [6].

Table 1 shows different attacks and the affected layers in the SDN architecture. The security vulnerabilities in SDNs are mainly distributed in application plane, control plane, and data plane. For instance, communication channels between isolated planes might be attacked by other compromised planes. Because of the visible nature of the control plane, it is more attractive to attacker and vulnerable to DoS and DDoS attacks [7]. The lack of authentication and authorization mechanisms is another security limitation in the application plane. The data plane is vulnerable to fraudulent flow rules, flow rule flooding, controller hijacking, and man-in-the-middle attacks.

Many security solutions, targeted at the exposed security vulnerabilities of SDN, are proposed. FRESCO [8] is proposed to enable development of OpenFlow security applications. The control plane security solutions consist of defending malicious applications, optimizing load balancing policies, DoS/DDoS attack mitigation, and reliable controller placement. In order to provide an ideally secure SDN control layer, Security-Enhanced (SE) Floodlight [9] adds a secure programmable northbound API to operate as an intercessor between applications and the data plane. The fine-grained security enforcement mechanisms such as authentication and authorization are used for applications that can change the flow rules. FortNOX [10] is a platform that enables the NOX controller to check flow rule contradictions real-time and authorize applications before they can change the flow rules.

However, rather than focusing on specific types of attacks or planes, it is desirable to develop and deploy security countermeasures in a more general manner to guarantee data and network resource security. Besides, how to systematically test and verify the deployed security mechanisms is another challenge. The current hardware deployment or software simulation are mostly focused on network function implementation and verification. None of them target the security perspective.

In addition, SDN switch hardware is too expensive to deploy massively and it is also difficult to modify the embedded protocols. Furthermore, commercial SDN switches cannot support customized protocols and interfaces. The software simulation (e.g., Mininet [11], NS3 [12]) is able to support large size SDN network topology but it lacks various hardware interfaces to enable hardware-based security schemes, e.g., TPM, special encryption chip, and random number generator.

To address the above problems, a lightweight security development platform—that supports security countermeasure implementation, deployment, and testing in a uniform, extensible, and economical mode—is desired. In this paper, we propose OSCO (Open Security-enhanced Compatible OpenFlow) platform, a uniform testing platform based on Raspberry Pi Single Board Computer (SBC) hardware and SDN network architecture, which supports highly configurable cryptographic algorithm modules, security protocols, flexible hardware extensions, and virtualized SDN networks. (An earlier version of this manuscript was presented in International Conference on Wireless Algorithms, Systems, and Applications (WASA), 2018 [13].) We systematically evaluate critical security modules, which include hash functions, stream/block ciphers, public-key cryptosystems, and key exchange protocols. Furthermore, we verify the scalability of the platform and the corresponding system overhead.

Contributions. In summary, we make the following contributions in this paper:

(i) We proposed an extensible security-oriented SDN network experiment platform, OSCO, allowing various security design schemes to experiment in SDN environment. Our scheme provides an open framework with flexible hardware interfaces and extensible

software modules that fully support standard OpenFlow protocol and its security enhancement solutions.

(ii) We implemented the OSCO platform in a Raspberry Pi hardware, which is acting as the OpenFlow switch. It supports physical SDN network deployment and simulation. In addition, we present a case study for switch port security enhancement to illustrate the extensibility and usability of OSCO platform functions with relatively low computational (extra 2.5% system overhead when performing AES-256 and SHA-256 functions) and networking performance overheads (73.7 Mb/s TCP and 81.2Mb/s UDP transmission speeds in 100Mb/s network settings).

Paper Organization. The rest of this paper is organized as follows: Section 2 presents the overall architecture and critical components of our platform with implementation details; Section 3 describes the case study for port security enhancement; Section 4 evaluates performance of some critical security modules; Section 5 discusses related work; and Section 6 concludes the paper.

2. Overall Architecture of OSCO Platform

In this section, we present the overall architecture of OSCO platform and describe its core functions with auxiliary components in detail.

2.1. Security Threats in SDN. As we discussed previously, the security threats of SDN exist in the application layer, control layer, and data layer, respectively, as well as the control-data-link and application-control-link interfaces. The application plane enables SDN applications to manipulate the behavior of SDN devices through the SDN controller. The variety of SDN applications allows the SDN network to be managed in the flexible and diverse ways via the centralized SDN controller. Because most of the network functions can be implemented as SDN applications, malicious applications can spread devastation in the whole network as well, specifically, attacks on authentication, access control mechanisms [6, 14–16], etc.

The control plane holds the abstract view of the entire network and provides the information of network resources to SDN applications. The control plane is implemented as a logically centralized module which separates the SDN application and data plane. Concrete implementation of control plane is the Network Operation System (NOS), which collects network information via APIs to observe and control network. NOS provides a uniform and centralized programmatic interface to the entire network. For instance, the controller is responsible for flow formation, management, and distribution in data-path elements via OpenFlow protocol between control and data plane. As the control plane is a centralized control entity, it is an attractive target to attackers. The main security challenges and threats existing in the control plane are unauthorized application access, e.g., DoS/DDoS attacks.

The data plane holds the SDN forwarding devices such as routers, switches, virtual switches, and access points and manages configurable device entity using OpenFlow protocol to control plane. The SDN device can be (re)configured for different purposes, including traffic isolation, data-path modification, and device virtualization via a remote procedure call from the SDN controller using an optional secure communication channel in data-control link. The flow rules are installed in the OpenFlow switch's flow tables according to the controller's decisions. The security challenges of data plane are compromising SDN controller and recognizing genuine flow rules and distinguishing them from bogus rules. And it is also vulnerable to saturation attacks, man-in-the-middle attack, and TCP-Level attacks.

2.2. Objectives. The goal of our proposed OSCO platform is to provide a uniform, open, extensible, and economical environment for security countermeasure implementation, deployment, and test in different SDN planes. In the application plane, OSCO is able to support deployment of SDN applications and its corresponding security countermeasures for authentication, authorization and access control, and accountability security risks. In the control plane, OSCO is capable of supporting various security-enhanced controllers via a standard interface to prevent unauthorized application access, DoS/DDoS attacks, etc. In the data plane, it functions as an OpenFlow compatible switch that forwards packages between switches and hosts, and plays an important role in the security countermeasure for saturation attacks, man-in-the-middle attacks, and TCP-Level attacks. OSCO is a testing environment for security schemes in all the three SDN planes.

2.3. Architecture of OSCO Platform. Figure 2 shows the overall architecture of the OSCO platform, which consists of central part enclosed by the dashed rectangle and peripheral system. The central part consists of OSCO core function modules. The peripheral system consists of the REST-API formatted SDN security applications, diversified peripheral security hardware, SDN controllers, and the security-supported network simulator. The OSCO central part functions as an OpenFlow switch by using embedded OpenFlow protocol under the SDN architecture. The SDN controller communicates with applications and OSCO core. The OSCO is able to extend the security function through the peripheral security hardware. The SDN network simulator is available for OSCO platform to simulate a large scale complex SDN network topology with an internal or external SDN controller.

(a) OSCO Core Function Modules: The central part of the OSCO platform includes core function modules, such as the *hardware interface*, the *Linux kernel*, the *OpenFlow module*, the *cipher algorithm library*, and *protocol stack*. The core function modules act as a multifunction OpenFlow switch. The security hardware is supported by the hardware (HW) interfaces, such as the secure chip, random number generators, biological information acquisition, and recognition devices. The security hardware provides better encryption or decryption performance than software implementation. The OpenFlow module is implemented above the Linux kernel, which is one of the most important protocols in an SDN

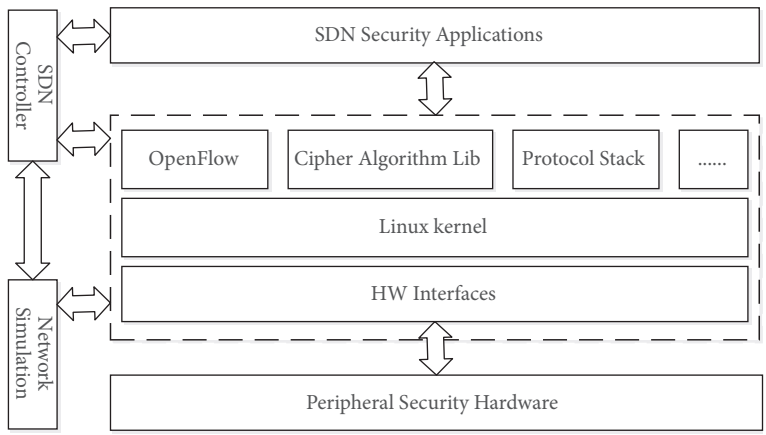


Figure 2: Overall architecture of OSCO platform.

architecture. OpenFlow is used for the interaction between a data plane constituted network switch and a control plane constituted controller. The cipher algorithm library contains the mainstream cryptographic algorithms, e.g., hash function, stream ciphers, and public-key cryptosystems. It provides rich algorithms and the latest implementation from different libraries. The key management, key distribution, and authentication protocols are loaded in the protocol stack module. In addition, other extensible modules such as package capture and package analysis modules can be employed in OSCO platform.

(b) Peripheral System: The peripheral system in OSCO platform supports the core function modules to fulfill their designed functionalities. The peripheral systems include four parts: (1) the SDN security application module, which implements and invokes the security solutions; (2) the peripheral security hardware module which provides hardware-based security solutions; (3) the network simulation module, which enables the simulation of the complex network topology and SDN network functions; (4) the SDN controller module, which makes packages forwarding decisions, maintains the SDN network topology, coordinates network resources, etc.

2.4. Implementation. (1) System Hardware: We select the Single Board Computer (SBC) hardware Raspberry Pi3 model B [17] as computation and HW-interface platform used in core modules implementation. The Raspberry Pi is an open source hardware with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor and 1 GB of RAM. It supports 10/100 Mbit/s Ethernet, 802.11n wireless, Bluetooth 4.1 On-board network, 4 USB2.0 ports, and 17 general purpose input-output (GPIO) connectors including I2C, SPI, UART, PCM, and PWM interfaces. We set up three external USB Ethernet adapters (10/100Mb/s) as OSCO physical ports. The Ubuntu MATE [18], which is a Linux-based implementation, is chosen as platform operation system (OS). The rest of the core functions modules are deployed above the Linux level. The Open Virtual Switch (OVS) [19] is installed as the implementation of OpenFlow protocol. OpenSSL [20], Crypto++ [21], and PBC [22] libraries are integrated as the cryptology algorithm

Figure 3: The hardware environment of OSCO platform.

module in the platform. Additionally, we use the Wireshark [23] as OpenFlow packet inspector and monitor. Figure 3 presents the hardware environment of our OSCO platform.

(2) Network Configuration: We implemented the POX/NOX [24] and Floodlight [9] SDN controllers in our platform, and a Floodlight controller based topology is shown in Figure 4. We chose Mininet [11] to create/simulate the SDN virtual network infrastructure. We discussed the implementation details of security enhancement based on OSCO platform by using a case study in Section 3.

3. Case Study: Port Security Enhancement on OSCO

3.1. System Configuration. As shown in Figure 4, in our case study, the experiment system consists of two parts, a virtual network and a physical network. We use Mininet to simulate a virtual SDN network with two switches and two hosts on the workstation (Intel i7-4770 CPU, 8G RAM, Ubuntu 16.04 64-bit OS). The host1 connects to the switch1, the host2 connects to the switch2, and the switch1 connects to the switch2. The Mininet configuration allows switch2 to connect to the controller (installing on the workstation with Intel i5-4570 CPU, 4G RAM, Ubuntu 16.04 64-bit OS) via a physical OSCO

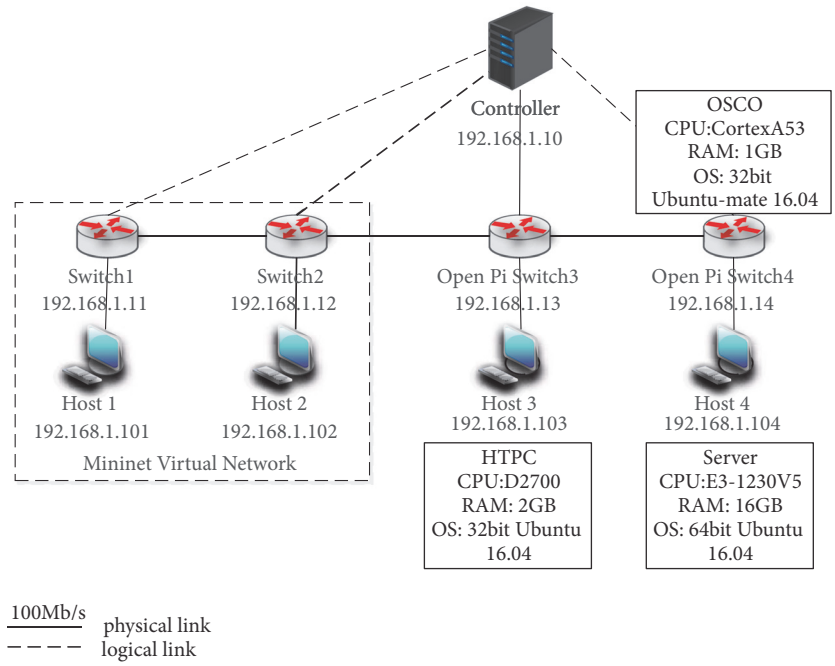


Figure 4: The system configuration.

switch (Cortex-A53 CPU, 1G RAM, Ubuntu MATE 16.04 32-bit OS).

The OSCO switch3 physically connects to OSCO switch4, the virtual switch2, and a controller. The host3 connects to the OSCO switch3, and the host4 connects to the OSCO switch4. The host3 is a HTPC with Intel Atom D2700 CPU, 2G RAM, Ubuntu 16.04 32-bit OS and the host4 is a server with Intel E3-1230v5 CPU, 16G RAM, Ubuntu 16.04 64-bit OS. The OSCO switch includes one original Ethernet port from Raspberry Pi and three USB Ethernet ports. The unified physical Ethernet interfaces are ensured by the core function modules of the platform.

OSCO switch runs OVS to support OpenFlow protocol and logical and physical ports mapping for OpenFlow connection. All switches connect (physically or logically) the controller via the OSCO switch3 and switches work in OpenFlow-only mode with OpenFlow 1.3 protocol.

3.2. Port Security Enhancement. The OVS software module is concrete standard implementation of OpenFlow protocol. It maps the physical port to the OpenFlow logical forwarding port and supports Spanning Tree Protocol to prevent broadcast storm in the loop network. For instance, in the implementation topology shown in Figure 4, the OVS port os1-eth1 in OSCO switch3 is used to connect os2-eth0 port in OSCO switch4 by connecting two physical ports with setting OVS bridge name and port mapping at both sides. The openness of the OVS makes it easier to extend the OpenFlow protocol. The OpenSSL library is used to add OSCO switch port encryption/decryption function by modifying OVS software module.

OpenFlow protocol processes packets through flow tables and actions in the OpenFlow pipeline. OpenFlow port receives and sends data to controller or switches. The flow tables hold the package forwarding rules and the actions apply the rules that include flow modification, deletion, addition, and forwarding. The process of the OpenFlow packet receiving and forwarding is conducted in the OpenFlow pipeline.

In this case study, one of the requirements is to enable OSCO switch to execute encryption/decryption operations in a specific port. To achieve this, the source code of OpenFlow pipeline part needs to be modified to enable os1-eth1 and os2-eth0 ports to perform packages encryption/decryption actions between them. The datapath.c file located in the OVS project data-path directory holds the implementation of OpenFlow pipeline. It contains the most important function calls and processing logic for the package forwarding. Pseudocode 1 presents the OVS original code and the key part of modified code for the implementation of port encryption.

Pseudocode 2 list is developed for package encryption, which is put in line 27 before the dp_xmit_skb function in original code.

The dp_output_put function processes the flow packages and sends them to the corresponding ports, which include the flood port, table port, controller port, and physical port. Only the os1-eth1 and os2-eth0 port encrypt or decrypt the packages. The check_port function checks the output port before a series of encryption efforts. The package needs to attach its hash value for the later integrity verification. The do_hash function calculates the package hash value by using SHA-256 implementation in OpenSSL library. The add_md_to_tail function adds the calculated hash value to the end of the package. To facilitate the implementation and evaluation, the encryption key and the initialization value are hard code inside the do_encrypt_data function.

```
1 int dp_output_port
2   (struct datapath *dp, struct sk_buff *skb, int out_port, int ignore_no_fwd)
3 {
4       BUG_ON(!skb);
5       switch (out_port){
6       ...
7       ...
8       case 0 ... DP_MAX_PORTS - 1: {
9          struct net_bridge_port *p = dp->ports[out_port];
10          if (p == NULL)
11             goto bad_port;
12          if (p->dev == skb->dev) {
13                /* To send to the input port, must use OFPP_IN_PORT */
14             kfree_skb(skb);
15             if (net_ratelimit())
16             printk(KERN_NOTICE "%s: can't directly "
17                 "forward to input port\n",
18                 dp->netdev->name);
19           return -EINVAL;
20           }
21          if (p->config & OFPPC_NO_FWD && !ignore_no_fwd) {
22            kfree_skb(skb);
23            return 0;
24           }
25           skb->dev = p->dev;
26
27           return dp_xmit_skb(skb);
28                    }
29
30        default:
31           goto bad_port;
32       }
33       ...
34       ...
35 }
```

Pseudocode 1:

```
1 if (chech_port (out_port,"os1-eth1","os2-eth0")    {
2             int32_t p_size = get_ksb_package_size (skb);
3             unsigned char package [p_size] = get_package_data (skb);
4
5             unsigned char md[33] = {0};
6             do_hash (package, p_size, md); // do SHA256
7             add_md_to_tail (md, skb); // to add hash value
8
9             int32_t size = get_data_size (skb);
10            unsigned char data [size] = get_data (skb); // get data from skb
11
12            int32_t padding_size =0;
13            encrypt_buff = (unsigned char *)malloc (padding_size);
14            do_encrypt_data (data, &encrypt_buff, padding_size);
15            put_encypted_data (encrypt_buff, skb);
16            free (encrypt_buff);
17 }
```

Pseudocode 2:

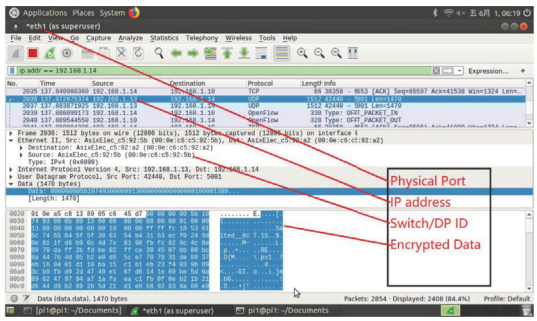


Figure 5: Package capturing in port os1-eth1.

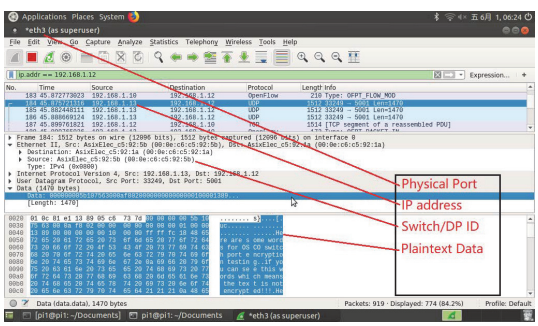


Figure 6: Package capturing in port os1-eth3.

The do_encrypt_data function encrypts data via invoking OpenSSL library to implement AES-CBC algorithm with 256-bits key size. Finally, the new package with encrypted data is sent to output port buffer via the dp_xmit_skb function.

Pseudocode 3 indicates the do_port_input function that handles the package receiving process in the OVS original code. The decryption action should be executed at line 10 before fwd_port_input function. The decryption action code is shown in Pseudocode 4.

The do_port_input function is responsible for retrieving the packages from input port in OpenFlow pipeline. The encrypted packages need to be decrypted before any further processing. It needs to check the input port before performing decryption action, because it only encrypts package that goes through os1-eth1 or os2-eth0 port. The get_data function gets a package from a network device buffer, and the do_decrypt_data function decrypts package with hard coded encryption/decryption key and initialized value. Next, it needs to calculate a hash value of decrypted package via calling do_hash function. The check_not_the_same_md function verifies the integrity of decrypted package by comparing two hash values. If two hash values are equal, it will call fwd_port_input function for the further package processing; otherwise it drops this package. The port encryption/decryption function is only available between the connection of OSCO switch3 and switch4 in our case. By doing this, the different block/stream encryption module can be easily tested and verified in OSCO platform.

Figures 5 and 6 represent the packages capturing on os1-eth1 and os1-eth3 ports in OSCO switch3. They show packages that contain some text data going through the physical port os1-eth1 and os1-eth3 in OSCO switch3, respectively.

Figure 5 indicates that the encrypted data is sent from OSCO switch3 to OSCO switch4 via port os1-eth1. 192.168.1.13 and 00:0e:c6:c5:92:5b are the IP address and the switch logical MAC address (it is also the ID of OpenFlow Data-Path) for OSCO switch3. The IP (192.168.1.14) and MAC (00:0e:c6:c5:92:a2) address are used by OSCO switch4.

As shown in Figure 6, the nonencrypted data (plaintext) are sent out from OSCO switch3 via port os1-eth3. The package will not be encrypted when using other port rather than the port os1-eth1.

All ports of OSCO switch3 are listed by SDN controller in Figure 7. It shows OSCO switch3's MAC address, IP address, OpenFlow version, and five registered ports in SDN controller. There are one OpenFlow logical port (local:os1) and four physical ports, but only one port (os1-eth1) allows encryption function. In addition, OpenFlow local port contains all physical ports, and its MAC address is the switch/OpenFlow Data-Path ID.

In this section, we present a case study for OSCO switch port security enhancement and illustrate the details to implement the port encryption/decryption function in OSCO switch. We evaluate the system and network performances in Section 4.

4. Performance Evaluation

In this section, we evaluate the OSCO platform in two aspects, computational overhead caused by the cryptographic security modules and its network performance. Table 2 indicates the hardware and software configuration of cryptographic algorithms performance testing.

We adopted the benchmark in OpenSSL project for the cryptographic algorithms performance testing, which is the most widely used in cryptographic evaluation. Besides Raspberry Pi-based OSCO platform, we also conduct the performance evaluation on two other hardware platforms deployed in our experiment (as shown in Figure 4). The Home Theatre PC (HTPC) has a duel core Intel Atom D2700, 2GB of RAM and running 32bit Ubuntu 16.04 OS. The Server platform has a quad-core Intel Xeon E3-1230 V5 processor with 16GB RAM and 64bit Ubuntu 16.04 OS installed. The experiment results disclose the hardware preference for different crypto settings. It could be a reference for SDN hardware selection according to the network and system performance requirements.

We mainly conducted the performance evaluation on four kinds of cryptographic operations, hash functions, symmetric encryption, public-key encryption, and digital signature, which are the basic primes of most SDN security solutions.

4.1. Hash Algorithm Performance. Hash function is used to ensure the integrity of transmitted data. We test performances of five widely used hash functions on three different hardware platforms. Table 3 shows the hash function parameters such

```
1 static void do_port_input (struct net_bridge_port *p, struct sk_buff *skb)
2 {
3         skb = skb_share_check (skb, GFP_ATOMIC);
4         if (!skb)
5             return;
6         /* Push the Ethernet header back on. */
7         skb_push (skb, ETH_HLEN);
8         skb_reset_mac_header (skb);
9
10        fwd_port_input (p->dp->chain, skb, p);
11 }
```

Pseudocode 3:

```
1 if (chech_bridge_ports (p,"os1-eth1","os2-eth0")    {
2
3             int32_t size = get_data_size (skb);
4             unsigned char data [size] = get_data (skb);
5             // get data from skb
6             int32_t padding_size =0;
7
8             decrypt_buff = (unsigned char *)malloc (padding_size);
9             do_decrypt_data (data, &decrypt_buff, padding_size);
10            // do AES CBC 256 decryption
11            put_decypted_data (decrypt_buff, skb);
12            free (decrypt_buff);
13
14            int32_t p_size = get_ksb_package_size (skb);
15            unsigned char package [p_size] = get_package_data (skb);
16
17            unsigned char md[33] = {0};
18            do_hash (package, p_size, md); // do SHA256
19            unsigned char oldmd = get_hash_value (skb, 256);
20
21            if (check_not_the_same_md (md, oldmd)) {
22                kfree_skb (skb);
23                return;
24            }
25 }
```

Pseudocode 4:

as output message size, maximum input message size, input processing block size, basic word size, and the number of processing steps.

Figure 8 shows the testing results of Message-Digest Algorithm (MD5) [25], Keyed-Hashing for Message Authentication (HMAC) [26], Secure Hash Algorithm (SHA) [27], and SHA-256 and SHA512 hash functions, which are executed by single thread with 16-byte input size in three platforms. The x-axis indicates the amount of processed data per second. The y-axis labels the five hash functions. The performance metric is defined as the parameter p_1, $p_1 = d_k/t$, where d_k denotes the size of data being processed with input buffer size k and t denotes the processing time.

For instance, we run MD5 hash routine in a loop for 3 seconds with a 16-byte input in OSCO platform. It performs over 1.7 million iterations after 3 seconds, that is, about 26.7 million bytes processed. Because of the use of Xeon E3 processor, the server has the best performance in three platforms; especially SHA1 has the best score in all five hash functions. The rest of the results also indicate that SHA1 is designed to perform better than MD5 when handling small input data size (16 bytes). The Server platform is the only one which runs 64-bit implementation of OpenSSL. Because of less number of steps, SHA-256 is the fastest hash function in both HTPC and OSCO platforms when handling 16-byte input data in 32-bit OS. HMAC generates a message authentication code by performing an embedded hash function. Its performance is close to embedded hash function (MD5 in our case). Figures 9, 10, and 11 show the hash function performances with different input data sizes

Table 2: Testing environment of cryptographic performance.

Platform	OSCO	Server	HTPC
CPU	Quad-core ARM Cortex-A53	Quad-core Intel Xeon E3-1230 V5	Quel core Intel Atom D2700
RAM	1GB	16GB	2GB
OS	32Bit Ubuntu-mate 16.04	64Bit Ubuntu 16.06	32Bit Ubuntu 16.04
Benchmark	OpenSSL 1.0.2g	OpenSSL 1.0.2g	OpenSSL 1.0.2g

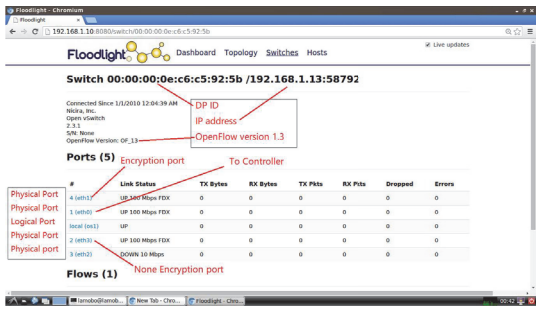


Figure 7: OSCO switch3 ports in SDN controller.

Table 3: Hash functions parameters.

Algorithm	MD5	SHA1	SHA256	SHA512
Message digest size (bits)	128	160	256	512
Message size (bits)	unlimited	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$
Block size (bits)	512	512	512	1024
Word size (bits)	32	32	32	64
Number of steps	80	80	64	80

on the three platforms, respectively. The x-axis indicates the different size of input data. The y-axis shows the amount of processed data per second (same as Figures 10, 11, 13, 15, and 14). According to Figure 9, on the OSCO platform, the performance is influenced by the input size, that is, the bigger the input size, the better the performance.

Furthermore, MD5 is somewhat less CPU-intensive than SHA1 in 32-bit OS [28]. The result shows MD5 getting better performance with input data larger than 64 Bytes (512 Bits) in Figures 9 and 10, because the additional block size padding operations reduce when input data size is larger than block size (it means larger than 512 bits). The similar result (large input size with better results) is presented in Figure 11, which is acquired from the Server platform. The 64-bit SHA-512 implementation is the only function among those four which benefits from a Streaming SIMD Extensions 2 (SSE2, Intel processor supplementary instruction sets) implementation [28]. Thus, it explains SHA512 performing better than SHA-256 in Server platform.

4.2. Stream/Block Ciphers Performance. Stream/block ciphers operate as important elementary components in many cryptographic protocols and are widely used to implement encryption of bulk data, for instance, RC4 [29], Blowfish [30], International Data Encryption Algorithm (IDEA) [31], Data Encryption Standard (DES) [32], and Advanced Encryption Standard (AES) [33] stream/block ciphers algorithms.

In this section, we evaluated the implementation performances of Blowfish, IDEA, DES, Triple DES (3DES), and AES in the Cipher Block Chaining (CBC) mode. Table 4 shows the block cipher parameters such as block size, key size, and round of iterations.

Figure 12 indicates that the RC4 is the fastest algorithm of all five algorithms in three platforms with single thread and 16-byte input size. RC4 is stream cipher; it simply uses bitwise exclusive-OR (XOR) operation between pseudorandom key-stream and plaintext stream for encryption. The decryption requires the use of the same key-stream and XOR operation. The RC4 only needs the hardware to perform XOR operation and to generate a pseudorandom key-stream. The simplest operations and less system hardware overhead guarantee the highest execution efficiency. The x-axis indicates the amount of processed data per second. The y-axis shows the type of algorithms.

DES is the deprecated data encryption standard. Its key size is 56 bits long, which is too short for proper security. 3DES is a security-enhanced version of DES. The 3DES module uses Encryption-Decryption-Encryption mode implementation, which applies two keys (k_1, k_2, 56-bits for each key) for each data block. 3DES provides a relatively simple method of increasing the key size of DES to protect against brute-force attack, but it increases the system overhead and reduces the speed of encryption and decryption. 3DES is the slowest algorithm of all cryptographic modules on all three platforms.

The DES has been superseded by the AES. AES supports 128, 192, or 256-bit key size with 128-bit block size. Recently, new processor hardware has been optimized for AES and SHA algorithms. For example, Intel's AES-NI and ARM's ARMv8-A instruction sets can provide faster hardware acceleration when using AES encryption. Intel XEON E3-1230 V5 and ARM Cortex-A53 processors support AES-NI and ARMv8-A instruction set, respectively. This explains why AES has better performance than DES in both server and OSCO platforms as shown in Figures 13 and 14.

Blowfish and IDEA are two commonly used symmetric-key block cipher algorithms with 64-bit data block size and 128-bit key size in our implementation. Blowfish has a variable key length from 32 bits up to 448 bits and employs 16-round interactions with key-dependent S-boxes. Blowfish provides a good encryption rate in software implementation on three platforms. IDEA consists of a series of 8 identical transformations and an output transformation, and the processes for encryption and decryption are similar. Blowfish

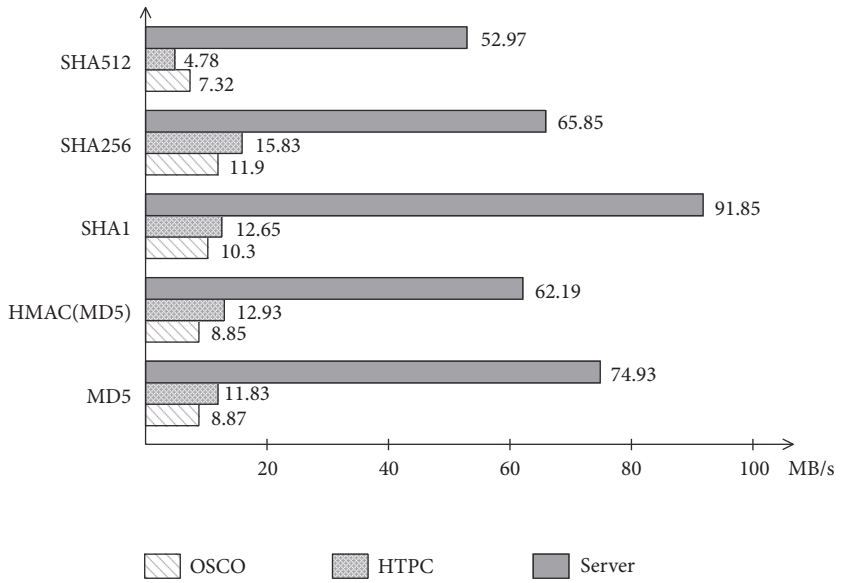


Figure 8: Hash functions performance in the three platforms.

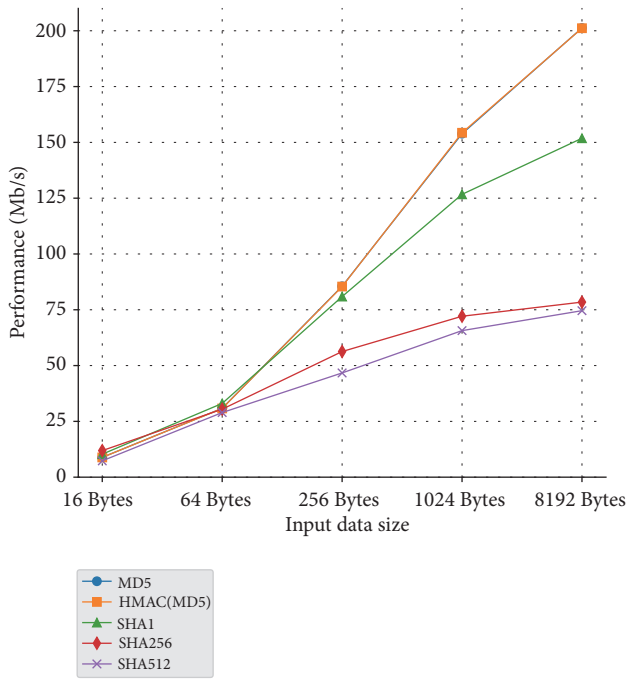


Figure 9: Hash functions performance in OSCO platform.

Table 4: Block cipher parameters.

Algorithm	Blowfish	IDEA	DES	3DES	AES-128	AES-192	AES-256
Block size (bits)	64	64	64	64	128	128	128
Key size (bits)	128	128	56	112	128	192	256
Rounds	16	8	16	48	10	12	14

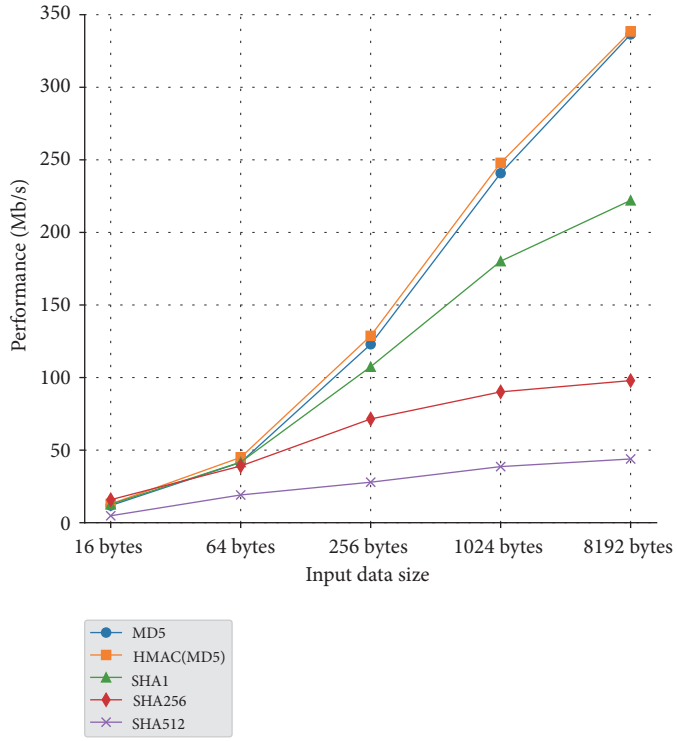


Figure 10: Hash functions performance in HTPC platform.

Table 5: Key parameters used in the experiment.

Algorithm	Modules size or size of n in bits
RSA	2048, 4096
DSA	1024, 2048
ECDSA	192, 224, 256, 384, 521
ECDH	192, 224, 256, 384, 521

and IDEA both show much better performance than DES in OSCO and Server platform.

HTPC platform has an old Atom D2700 processor which cannot support any new security instruction set such as AES-IN or Intel SHA extensions. This is the reason that AES is slower than DES in HTPC platform, as shown in Figure 15.

The performance of IDEA is very similar to AES-128 on HTPC platform. Blowfish also shows a good performance (the second best performance) on HTPC platform. In particular, longer key size leads to lower performance in AES. The performance of block cipher algorithm (Blowfish, IDEA, DES, 3DES, and AES) is not significantly affected by the size of input data on the three platforms.

4.3. Public-Key Cryptosystems Performance. The public-key cryptosystems are widely used for key distribution, confidentiality, and authentication. Table 5 shows the key parameters used by the algorithm in the experiment.

RSA (Rivest-Shamir-Adleman) [55] is one of the first practical public-key cryptosystems. In the RSA scheme, the parameter n is the block size and also participates in generating the public key and the private key. A typical size for n is 1024 bits. In our experiment, the length of the parameter n is set as 2048 bits and 4096 bits.

As a commonly used digital signature algorithm, DSA scheme [56] provides similar performance in signing and verifying operation as shown in Figures 16 and 17. The signing and verifying operation performance metrics are defined as $p_2 = O_s/t$ and $p_3 = O_v/t$, where O_s and O_v denote the number of executed signing operations and verification operations, respectively, and t denotes the time cost of the operation.

Figure 16 shows that ECDSA-192 [57] has the best signing result and RSA-4096 has the worst signing operation in all platforms. The RSA-2048 has the best verification performance and ECDSA-521 shows the worst performance in verification in Figure 17. The Server platform is the fastest one running both signing and verifying operations. The OSCO platform is better than HTPC platform in both operations. The x-axis indicates the amount of accomplished operations per second. The y-axis represents the type of algorithms with different key size (same as Figures 16, 17, and 18).

In our platform, we did not deploy the classic textbook RSA scheme due to its insecure features. Instead, our implementation supports the probabilistic encryption by using the RSA-PKCS1-PADDING padding mode to satisfy the desired security requirements. The RSA-PKCS1-PADDING padding mode adds random nonzero data to each data block to ensure that the result differs each time. The similar technique is used by other algorithms such as DSA and ECDSA.

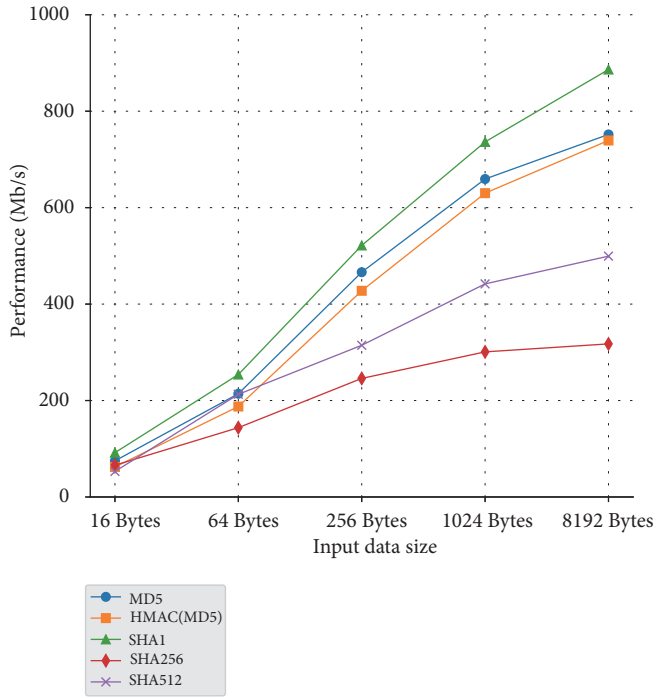


Figure 11: Hash functions performance in Server platform.

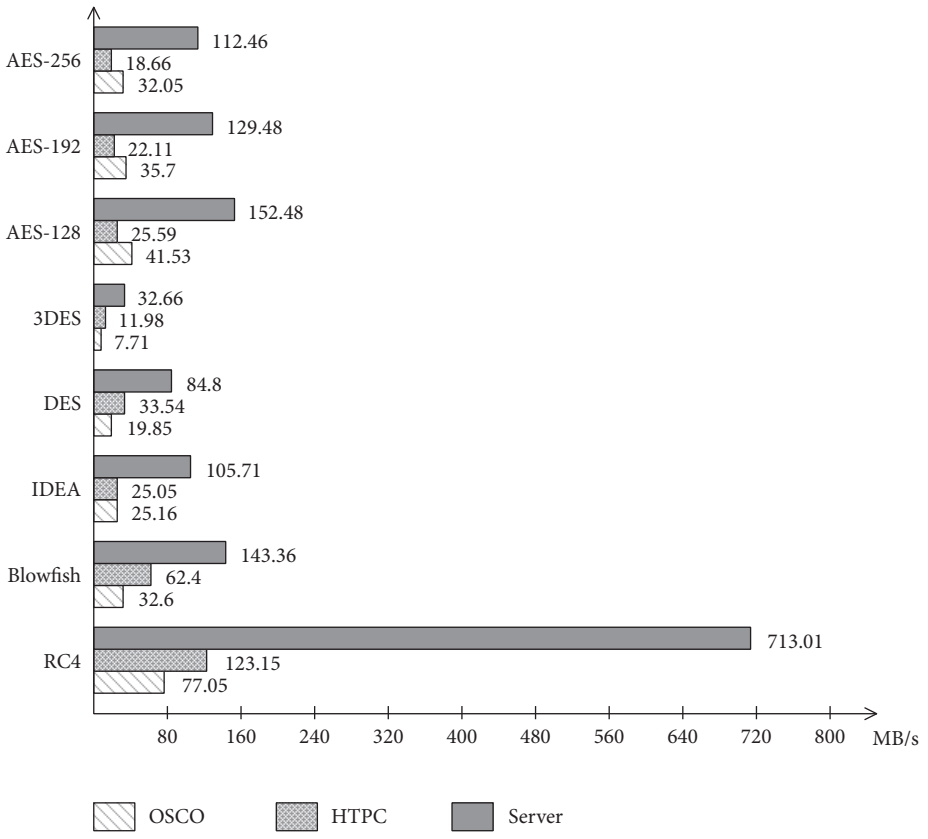


Figure 12: Stream/block ciphers performance in the three platforms.

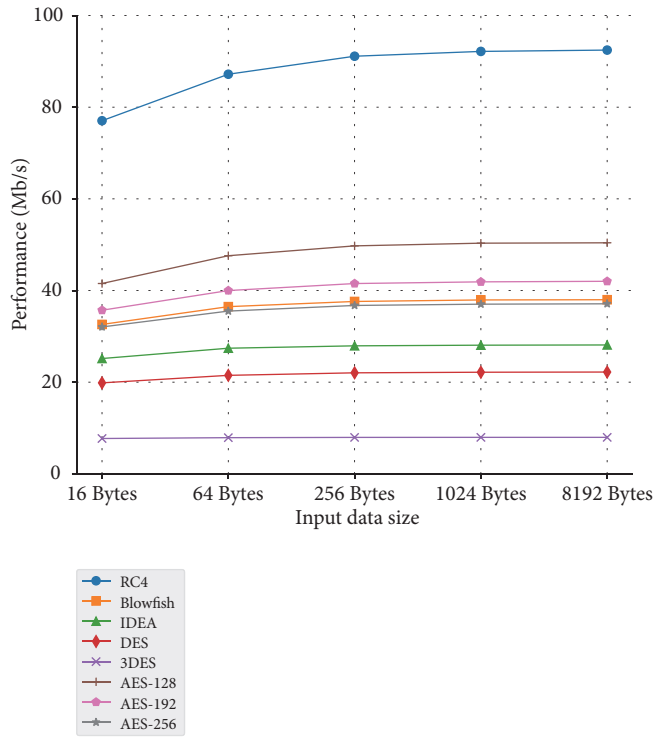


Figure 13: Stream/block ciphers performance in OSCO platform.

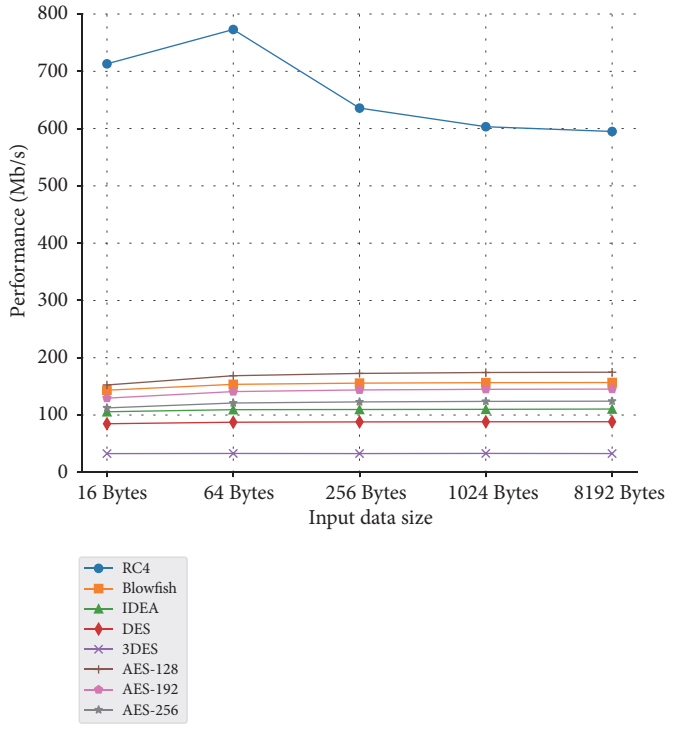


Figure 14: Stream/block ciphers performance in the Server platform.

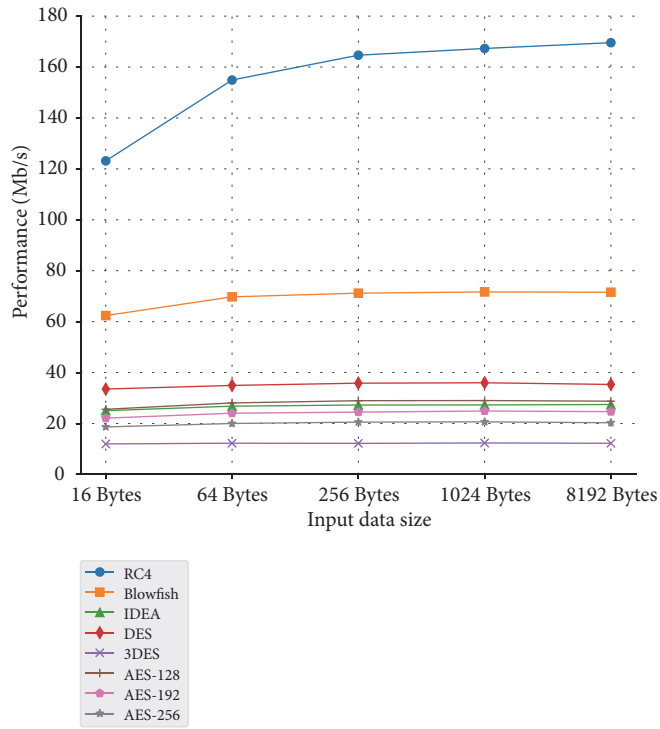


Figure 15: Stream/block ciphers performance in HTPC platform.

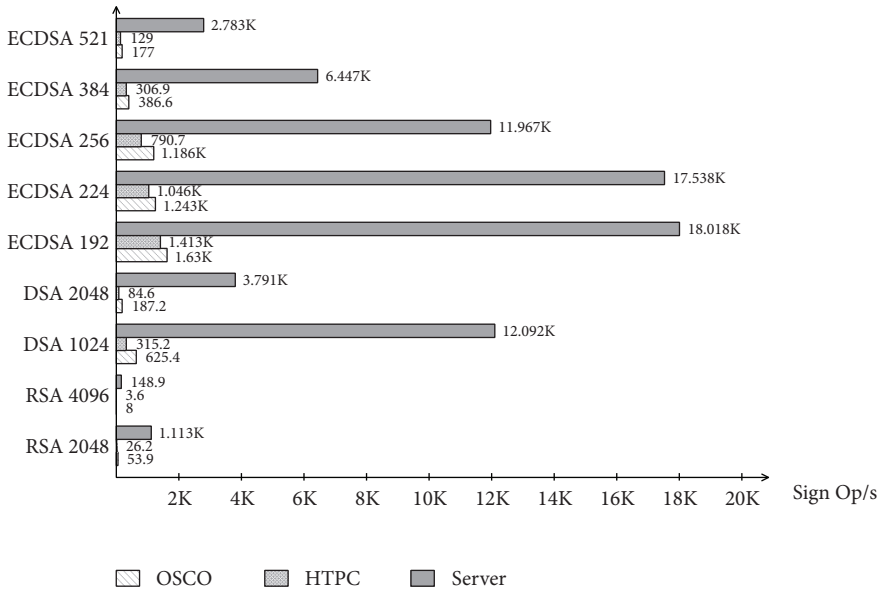


Figure 16: Sign operation performance in the three platforms.

4.4. ECDH Key Exchange Performance. The ECDH (elliptic-curve Diffie-Hellman) [58] is a key agreement protocol that allows two parties to establish a shared secret over an insecure channel. The ECDH key exchange performance is influenced by the size of key as shown in Figure 18. ECDH-192 has the best performance in OSCO platform, while on the Server platform, ECDH-224 is much faster than ECDH-192. ECDH and ECDSA-224/256 implementation are optimized in 64-bit OpenSSL after version 1.0.0h [59]. ECDH-192 and ECDSA-192 only have 32-bit portable implementation. The ECDH operation performance metrics are defined by $p_4 = O_i/t$, where O_i denotes the number of executed ECDH operations with key size i and t denoting the time cost of the operation. The benchmark runs a standard NIST (National

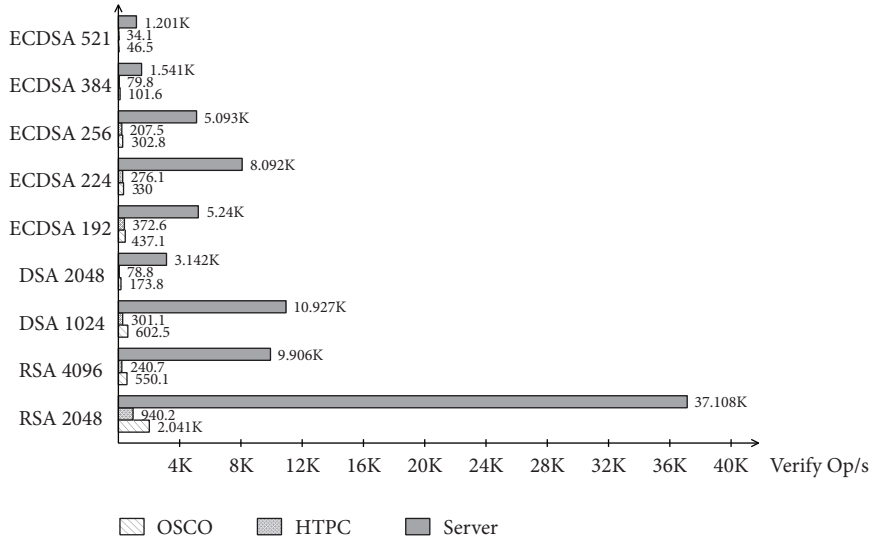


Figure 17: Verify operation performance in three platforms.

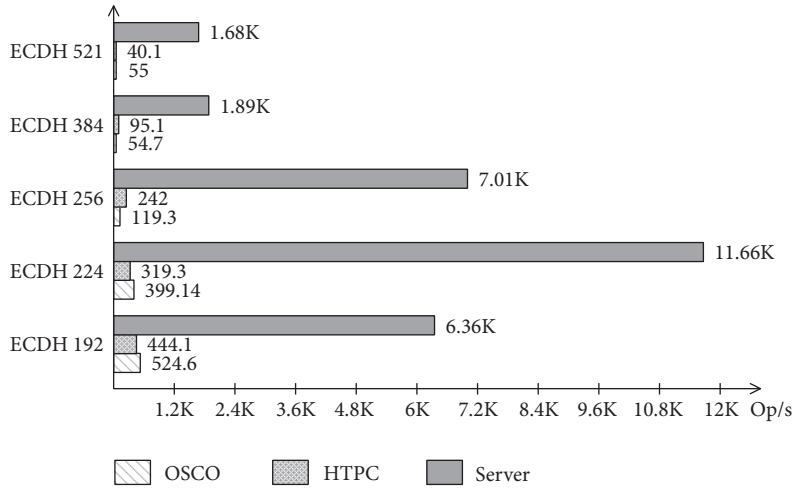


Figure 18: ECDH key exchange performance in the three platforms.

Institute of Standards and Technology) implementation of elliptic-curve with different key lengths, such as P-192, P-224, P-256, P-384, and P-521.

4.5. OSCO Platform Network Performance. The OSCO acts as an OpenFlow switch in the SDN network topology with some enhanced security abilities. Its data forwarding performance, which influences the whole SDN network performance, is one of the key factors besides the computation power of the cryptosystem.

The experiment uses IPerf, a Linux network tool, to test the max bandwidth and minimum response time of two hosts, which use the TCP and UDP network connection, respectively. The UDP connection testing applies -ub 100m as the input parameters and the other part keeps using the default setting; e.g., UDP buffer size is 160Kbyte and port is 5001. The TCP connection uses all default parameters for setting; e.g., TCP windows is 43.8Kbyte, etc. The network performance metrics are defined by $p_5 = d_p/t$, where d_p denotes the number of data being transmitted under protocol p mode and t denotes the transmission time.

Figure 19 shows two different network connections in TCP and UDP mode. The first connection (the virtual host1 to the physical host3) is indicated by lined bar, and the second connection is the link between two physical hosts (host4 to host3), indicated by dotted bar (the network topology is shown in Figure 4). The solid bar indicates the maximum theoretical network connection bandwidth (100Mb/s). Both connections get pretty good results compared with maximum theoretical bandwidth. In addition, the minimum response time of the 1st connection is 7.902ms, and the 2nd connection gets 0.029ms (both use UDP connection).

Figure 20 compares network performances between two OSCO switches when they enable or disable real-time encryptions. Based on the reviewed results of different cryptographic algorithms, AES-256 and SHA-256 algorithms

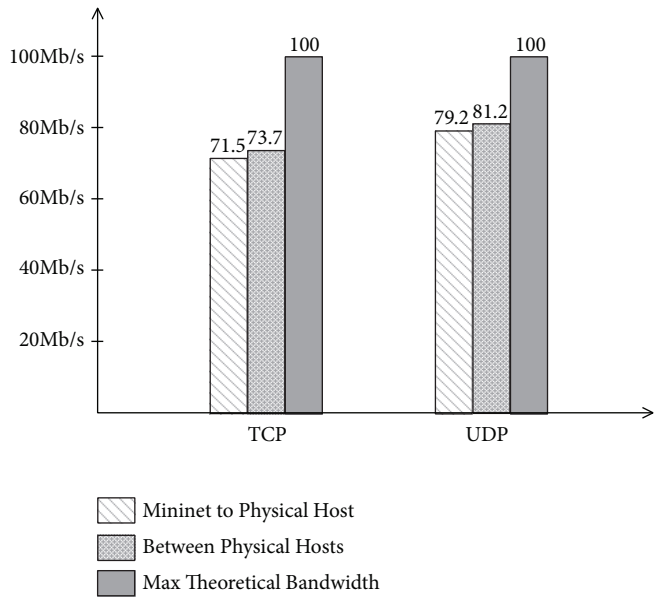


Figure 19: Network performance in OSCO platform.

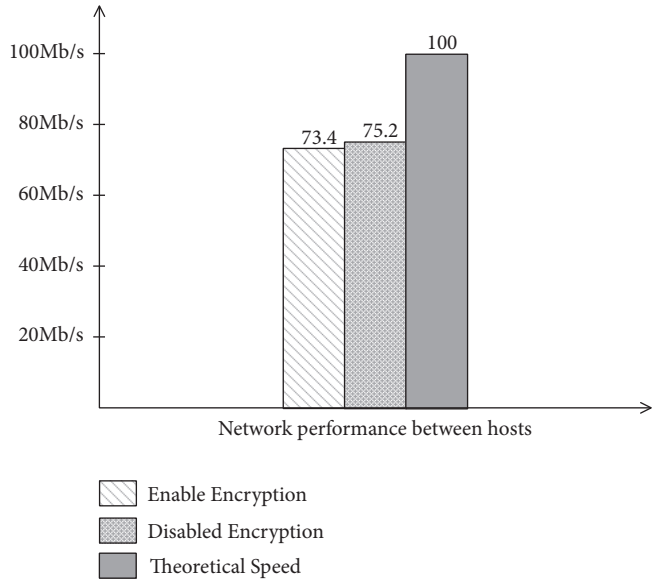


Figure 20: Network performance with security function enabled/disabled in OSCO.

are chosen to build a custom real-time encryption protocol on the OSCO platform. The experiment shows that the encryption overhead is relatively low (around 2.5%) via a UDP connection between two switches.

In this section, we evaluate the performance of most used cryptographic algorithms in different platforms. Moreover, our experiments confirm that OSCO platform is not only capable of carrying out the various cryptographic algorithms but also able to support good network bandwidth and speed requirements. Furthermore, the OSCO platform has the lowest power consumption of 4W (system power consumption) comparing to HTPC and Server platform which approximately consume 15W and 80W only for the processors (Thermal Design Power from Intel), respectively.

5. Related Work

With the rapid development of SDN, more and more studies are carried out to improve SDN network performance and security. The centralized SDN architecture makes network management much more easier and flexible than before, but it introduces new security challenges too. Therefore, many security-oriented solutions are proposed. The proposed solutions focus on the security challenges in the data, controller,

Table 6: SDN security solutions.

Solutions	Type	SDN Plane			Interface	
		Data	Ctrl.	App.	App-Ctrl	Ctrl-Data
PERM-GUARD [15]	Cross-layer protection framework	x	x	x		
VeriFlow [34]	Verify and debug flow rules	x				
CPRecovery [35]	Controller response framework	x	x			
FortNOX [10]	Controller framework	x	x			
FlowChecker [36]	Configuration verification tool	x	x			
FRESCO [8]	Anomaly diction and mitigation framework		x		x	
PermOF [37]	Permission control system for Apps.	x	x			
Flover [38]	Flow policy verification	x	x	x		
OFTesting [39]	OF Apps testing and debugging			x		
SE-Floodlight [40]	Conflict resolution, authorization, audit system	x	x		x	
Mini-Cut placement [41]	Controller reliability, switch-controller connectivity	x	x			x
Monitoring [42]	Data plane connectivity monitoring	x				x
POCO [43]	Controller resilience and failure tolerance		x			x
DCP [44]	Dynamic controller provisioning, scalability, availability		x			x
Resonance [45]	Access control and dynamic policy enforcement	x				
AVANT-GUARD [46]	Controller plane security enhancement framework		x			

Table 7: Platform solutions.

Platform solutions	Software environment	Hardware environment
Mininet [11]	x	
NS3 [12]	x	
OpenNet [47]	x	
OMNET++ [48]	x	
EstiNet [49]	x	
Donatini et al. [50]		x
Ariman et al. [51]		x
OpenSample [52]		x
Mercury [53]		x
XSM [54]		x

and application plane of SDN as well as the interfaces in between. For instance, AVANT-GUARD [46] protects the control plane from a variety of attacks. The authors in [10] propose FortNOX, a mechanism which monitors the flow rules states from application plane to data plane in the SDN network. PERM-GUARD [15] protects all the three SND planes through management of flow rule production permissions. Gao et al. [60] proposed CPSTCS, a cyber-physical systems testbed based on cloud computing and SDN in Industry 4.0. Yoon et al. [61] proposed a fault-tolerant mechanism that responds to controller failures in CPS that have multiple controllers for resilient control of physical objects. Piedrahita et al. [62] proposed a solution that detects and responds automatically to sensor attacks and controller attacks in industrial network based on SDN. Han et al. [63] proposed a dynamic route strategy to decrease the delay and congestion in cyber-physical power system based on SDN. Table 6 shows more SDN security solutions. The software simulation or the physical device deployment are used to test and verify those proposed theories or prototypes.

Generally speaking, there are two ways to verify a newly designed protocol or mechanism in a computer network system. One is software simulation verification, which models a real-system or hypothetical situation on a computer software so that it can be studied to see how the system works. The other way for physical device deployment verification deploys a designed protocol or mechanism to a real hardware environment. Table 7 shows more platform related solutions for the networking system.

Software simulation. As the network environment of SDN is different from the traditional network, the challenge of SDN network simulation has drawn a great attention in recent years. Lantz et al. [11] proposed the Mininet, a lightweight tool simulating the SDN networks by using OS-level virtualization features, including processes and network namespaces to scale hundreds of SDN nodes. Riley et al. [12] introduced NS3, a powerful tool for network simulation. Chan et al. [47] proposed OpenNet, which is an open source simulator for wireless network simulation. OpenNet is a combination of Mininet and NS3 and it supports various SDN controllers by using Mininet and wireless modeling through NS3. OpenNet also supports wireless channel scan mechanism. Wang et al. [49] introduced EstiNet, an SDN network simulator, which enables external SDN controller to control simulated OpenFlow switches.

Physical device deployment. Donatini et al. [50] introduced an approach to develop a SDN network measurement platform by using NetFPGA under the LTE network infrastructure. The SDN implementation part is based on OpenFlow and OpenSketch [64] projects. Ariman et al. [51] implemented a real-time testbed for software-defined wireless networks (SDWN) by using Raspberry Pi as OpenFlow

(OF) switches. The implementation provides a practical development and testing environment for SDWNs. Suh et al. [52] proposed and implemented a sampling-based measurement platform which decreases the latency to gather accurate measurements of network load by using sFlow [65] samples. Hu et al. [66] proposed an SDN based big data platform for social TV analytics. It provides three main components, distributed date crawler, big data processing, and social TV analytics. Skowyra et al. [67] introduced an SDN-Enabled application verification platform which verifies composed applications and SDN system models to assure their safety, security, and performances.

Unlike the related work mentioned above, we do not just focus on simulating some specific SDN network functions. Instead, we are interested in exposing the new security challenges introduced by SDN deployment and verifying corresponding countermeasures on an extensible customized platform.

6. Conclusion

In this paper, to effectively integrate security countermeasures and systematically verify and test developed security mechanisms in the SDN environment, we propose a lightweight security enhancement and security testing platform, OSCO (Open Security-enhanced Compatible OpenFlow) platform. The design of OSCO is based on Raspberry Pi Single Board Computer (SBC) hardware and SDN network architecture, which supports highly configurable cryptographic algorithm modules, security protocols, flexible hardware extensions, and virtualized SDN networks. We prototyped our design and present a case study for port security enhancement to illustrate the details of the OSCO platform implementation and deployment. We evaluate the prototype system, and the experiment results show that our system conducted security functions with relatively low computational and networking performance overheads.

Data Availability

No data were used to support this study.

Disclosure

An earlier version of this manuscript was presented in "International Conference on Wireless Algorithms, Systems, and Applications (WASA) 2018" [13].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (No. 2017YFB1400700), the National Natural Science Foundation of China (No. 61402029, U17733115), the National Natural Science Foundation of China (No. 61379002, No. 61370190), and Beijing Natural Science Foundation (No. 4162020).

References

[1] HPE Marie Paule-Odini, "Sdn and nfv evolution towards 5g," 2017, http://https://sdn.ieee.org/newsletter/september-2017/sdn-and-nfv-evolution-towards-5g.

[2] R. Wang, J. Yan, D. Wu, H. Wang, and Q. Yang, "Knowledge-Centric Edge Computing Based on Virtualized D2D Communication Systems," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 32–38, 2018.

[3] N. McKeown, T. Anderson, H. Balakrishnan et al., "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[4] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: a comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[5] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2317–2346, 2015.

[6] J. Spooner and S. Y. Zhu, "A Review of Solutions for SDN-Exclusive Security Issues," *International Journal of Advanced Computer Science and Applications*, 2016.

[7] M. Kuerban, T. Yun, . Y. Qing, and P. David, "Flowsec: Dos attack mitigation strategy on sdn controller," in *Proceedings of the IEEE International Conference on Networking*, pp. 211-212, 2016.

[8] S. Shin, P. Porras, Y. Vinod, M. Fong, G. Gu, and T. Mabry, "Fresco: Modular composable security services for software-defined networks," *Proceedings of Network & Distributed Security Symposium*, 2013.

[9] "Floodlight controller," http://www.projectfloodlight.org/.

[10] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, "A security enforcement kernel for OpenFlow networks," in *Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks, HotSDN 2012*, pp. 121–126, Association for Computing Machinery, Helsinki, Finland, August 2012.

[11] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets '10)*, 19 pages, ACM, October 2010.

[12] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," *Modeling and Tools for Network Simulation*, pp. 15–34, 2010.

[13] H. Cheng, J. Liu, J. Mao, M. Wang, and J. Chen, "OSCO: An Open Security-Enhanced Compatible OpenFlow Platform," in *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications (WASA)*, Springer, 2018.

[14] R. Klöti, V. Kotronis, and P. Smith, "OpenFlow: A security analysis," in *Proceedings of the 2013 21st IEEE International Conference on Network Protocols, ICNP 2013*, pp. 1–6, Institute of Electrical and Electronics Engineers, Goettingen, Germany, 2013.

[15] M. Wang, J. Liu, J. Chen, X. Liu, and J. Mao, "Perm-guard: Authenticating the validity of flow rules in software defined networking," in *Proceedings of the IEEE International Conference on Cyber Security and Cloud Computing*, pp. 1–17, 2016 (Chinese).

[16] P. P. Naga, K. Fang Hao, and T. V. Lakshman, *Securing software defined networks via flow deflection*, 2014.

[17] "Raspberry pi hardware specification," 2011, http://www.raspberrypi.org/documentation/hardware/.

[18] "Ubuntu mate," http://ubuntu-mate.org/.

[19] "Open vswitch," http://www.openvswitch.org/.

[20] "Open ssl," http://www.openssl.org/.

[21] "Crypto++ library," http://www.cryptopp.com/.

[22] "Pairing-based cryptography library," http://crypto.stanford.edu/pbc/.

[23] "Wireshark," http://www.woreshark.org/.

[24] "Nox controller," http://www.noxrepo.org/.

[25] R. Rivest, "The MD5 Message-Digest Algorithm," *RFC*, vol. 473, no. 10, pp. 492-492, 1992.

[26] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," RFC2104, 1997.

[27] D. Eastlake 3rd and P. Jones, "Us secure hash algorithm 1 (sha1)," Technical Report, 2001.

[28] T. Pornin, "Is calculating an md5 hash less cpu intensive than sha family functions?" stackoverflow, https://stackoverflow.com/questions/2722943/.

[29] R. Thayer and K. Kaukonen, *A Stream Cipher Encryption Algorithm*, 1999.

[30] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (Blowfish)," in *Fast Software Encryption, Cambridge Security Workshop*, pp. 191–204, 1993.

[31] X. Lai and J. L. Massey, *A Proposal for a New Block Encryption Standard*, Springer Berlin Heidelberg, 1991.

[32] M. E. Smid and D. K. Branstad, "Data encryption standard: past and future," *Proceedings of the IEEE*, vol. 76, no. 5, pp. 550–559, 1988.

[33] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-The Advanced Encryption Standard*, Springer, Berlin, Germany, 2002.

[34] A. Khurshid, W. Zhou, M. Caesar, and P. B. Godfrey, "Veriflow: Verifying network-wide invariants in real time," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 467–472, 2012.

[35] P. Fonseca, R. Bennesby, E. Mota, and A. Passito, "A replication component for resilient OpenFlow-based networking," in *Proceedings of the 2012 IEEE Network Operations and Management Symposium, NOMS 2012*, pp. 933–939, USA, April 2012.

[36] E. Al-Shaer and S. Al-Haj, "FlowChecker: Configuration analysis and verification of federated OpenFlow infrastructures," in *Proceedings of the 3rd ACM Workshop on Assurable and Usable Security Configuration, SafeConfig '10*, pp. 37–44, USA, October 2010.

[37] X. Wen, Y. Chen, C. Hu, C. Shi, and Y. Wang, "Towards a secure controller platform for OpenFlow applications," in *Proceedings of the 2013 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, HotSDN 2013*, pp. 171-172, China, August 2013.

[38] S. Son, S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Model checking invariant security properties in OpenFlow," in *Proceedings of the 2013 IEEE International Conference on Communications, ICC 2013*, pp. 1974–1979, Hungary, June 2013.

[39] M. Canini, D. Kostic, R. Jennifer, and D. Venzano, "Automating the testing of openflow applications," in *Proceedings of the International Workshop on Rigorous Protocol Engineering*, 2011.

[40] "OpenFlow Sec Security Enhanced Floodlight," SRI International, https://www.sdxcentral.com/projects/openflow-sec-security-enhanced-floodlight/.

[41] Y. Zhang, N. Beheshti, and M. Tatipamula, "On resilience of split-architecture networks," in *Proceedings of the Global Telecommunications Conference*, pp. 1–6, 2012.

[42] J. Kempf, E. Bellagamba, A. Kern, D. Jocha, A. Takacs, and P. Sköldström, "Scalable fault management for OpenFlow," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, pp. 6606–6610, June 2012.

[43] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran-Gia, "Pareto-optimal resilient controller placement in SDN-based core networks," in *Proceedings of the 25th International Teletraffic Congress (ITC '13)*, pp. 1–9, September 2013.

[44] M. F. Bari, A. R. Roy, S. R. Chowdhury et al., "Dynamic controller provisioning in software defined networks," in *Proceedings of the 9th International Conference on Network and Service Management (CNSM '13)*, pp. 18–25, Zürich, Switzerland, October 2013.

[45] A. Nayak, A. Reimers, N. Feamster, and R. Clark, "Resonance: Dynamic access control for enterprise networks," in *Proceedings of the 1st Workshop: Research on Enterprise Networking, WREN 2009, Co-located with the 2009 SIGCOMM Conference, SIGCOMM'09*, pp. 11–18, Spain, August 2009.

[46] S. Shin, V. Yegneswaran, and P. Porras, "AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pp. 413–424, Berlin, Germany, 2013.

[47] M.-C. Chan, C. Chen, J.-X. Huang, T. Kuo, L.-H. Yen, and C.-C. Tseng, "OpenNet: A simulator for software-defined wireless local area network," in *Proceedings of the 2014 IEEE Wireless Communications and Networking Conference, WCNC 2014*, pp. 3332–3336, Turkey, April 2014.

[48] A. Varga, "The omnet++ discrete event simulation system," in *European Simulation Multiconference*, 2001.

[49] S.-Y. Wang, C.-L. Chou, and C.-M. Yang, "EstiNet openflow network simulator and emulator," *IEEE Communications Magazine*, vol. 51, no. 9, pp. 110–117, 2013.

[50] L. Donatini, R. G. Garroppo, S. Giordano et al., "Advances in LTE network monitoring: A step towards an SDN solution," in *Proceedings of the 2014 17th IEEE Mediterranean Electrotechnical Conference, MELECON 2014*, pp. 339–343, Lebanon, April 2014.

[51] M. Ariman, G. Secinti, M. Erel, and B. Canberk, "Software defined wireless network testbed using Raspberry Pi of switches with routing add-on," in *Proceedings of the IEEE Conference on Network Function Virtualization and Software Defined Network, NFV-SDN 2015*, pp. 20-21, USA, 2016.

[52] J. Suh, T. T. Kwon, C. Dixon, W. Felter, and J. Carter, "OpenSample: A low-latency, sampling-based measurement platform for commodity SDN," in *Proceedings of the 2014 IEEE 34th International Conference on Distributed Computing Systems, ICDCS 2014*, pp. 228–237, July 2014.

[53] K. Lorincz, B.-R. Chen, and G. W. Challen, "Mercury: a wearable sensor network platform for high-fidelity motion analysis," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 183–196, Berkeley, California, USA, November 2009.

[54] P. Dutta, M. Grimmer, A. Arora, and S. Bibykt, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *Proceedings of the 4th International*

Symposium on Information Processing in Sensor Networks, IPSN 2005, pp. 497–502, USA, April 2005.

[55] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[56] National Institute Of Standards, *Digital signature standard (dss)*, vol. 25, Federal Information Processing Standards Publication 186-2, 2000.

[57] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.

[58] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology-CRYPTO'85*, H. C. Williams, Ed., vol. 218 of *Lecture Notes in Computer Science*, pp. 417–426, Springer, 1986.

[59] Changelog, "Changes between 1.0.0h and 1.0.1," OpenSSL, https://www.openssl.org/news/changelog.html#x32/.

[60] H. Gao, Y. Peng, K. Jia, Z. Wen, and H. Li, "Cyber-Physical Systems Testbed Based on Cloud Computing and Software Defined Network," in *Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 337–340, 2016.

[61] S. Yoon, J. Lee, Y. Kim, S. Kim, and H. Lim, "Fast controller switching for fault-tolerant cyber-physical systems on software-defined networks," in *Proceedings of the 22nd IEEE Pacific Rim International Symposium on Dependable Computing, PRDC 2017*, pp. 211-212, New Zealand, January 2017.

[62] A. F. M. Piedrahita, V. Gaur, J. Giraldo, A. A. Cardenas, and S. J. Rueda, "Leveraging Software-Defined Networking for Incident Response in Industrial Control Systems," *IEEE Software*, vol. 35, no. 1, pp. 44–50, 2017.

[63] Y. Han, H. E. Yiqian, F. Lou, Y. Wang, and C. Guo, "Analysis and application of sdn based dynamic optimal route strategy for cyber layer in cascading failures of cyber-physical power system," *Power System Technology*, 2018.

[64] "Opensketch project," https://github.com/lavanyaj/opensketch.git.

[65] sflow, http://sflow.org/about/index.php/.

[66] H. Hu, Y. Wen, Y. Gao, T.-S. Chua, and X. Li, "Toward an SDN-enabled big data platform for social TV analytics," *IEEE Network*, vol. 29, no. 5, pp. 43–49, 2015.

[67] R. Skowyra, A. Lapets, A. Bestavros, and A. Kfoury, "A verification platform for SDN-enabled applications," in *Proceedings of the 2nd IEEE International Conference on Cloud Engineering, IC2E 2014*, pp. 337–342, USA, March 2014.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 6829326, 10 pages
https://doi.org/10.1155/2018/6829326

Research Article

RPAR: Location Privacy Preserving via Repartitioning Anonymous Region in Mobile Social Network

Jinquan Zhang,[1,2] Yanfeng Yuan,[1] Xiao Wang,[1] Lina Ni,[1,2,3] Jiguo Yu,[4,5,6] and Mengmeng Zhang[1]

[1]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2]*Shandong Province Key Laboratory of Wisdom Mine Information Technology, Shandong University of Science and Technology, Qingdao 266590, China*
[3]*Key Laboratory of the Ministry of Education for Embedded System and Service Computing, Tongji University, Shanghai 201804, China*
[4]*School of Information Science and Engineering, Qufu Normal University, Shandong 276826, China*
[5]*Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China*
[6]*Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan 250014, China*

Correspondence should be addressed to Lina Ni; nln2004@163.com

Received 11 June 2018; Revised 2 October 2018; Accepted 17 October 2018; Published 5 November 2018

Guest Editor: Yan Huo

Applying the proliferated location-based services (LBSs) to social networks has spawned mobile social network (MSN) services that allow users to discover potential friends around them. While enjoying the convenience of MSN services, the mobile users also are confronted with the risk of location disclosure, which is a severe privacy preserving concern. In this paper, we focus on the problem of location privacy preserving in MSN. Particularly, we propose a repartitioning anonymous region for location privacy preserving (RPAR) scheme based on the central anonymous location which minimizes the traffic between the anonymous server and the LBS server while protecting the privacy of the user location. Furthermore, our scheme enables the users to get more accurate query results, thus improving the quality of the location service. Simulation results show that our proposed scheme can effectively reduce the area of anonymous regions and minimize the traffic.

1. Introduction

Internet of Things (IoT), a trend of future networks, is immersed into many aspects of our personal and working lives and provides more comprehensive intelligent service. Social networks widely used in mobile Internet catalyze mobile social networks (MSNs), and users in MSN can not only acquire their own location information and sign in a location but also find nearby friends and access to location-based services (LBSs) such as finding the nearest hotel, finding directions, sharing action tracks, obtaining information of body area networks, and so on [1–7]. However, when we enjoy the convenience of LBS and MSN services, the mobile users also are confronted with the risk of location disclosure, which is a severe privacy preserving concern [8–12].

MSN services (MSNS) have a wide range of applications in people's daily life, where location-aware information plays a very important role [13]. In addition to supporting real-time services, location-based applications can predict the behavior of the users by analyzing the user's position traces to obtain the user's interest preferences and make user interest recommendations. With the development of MSNS, location privacy issues are gradually attracting more and more people's attention. The location information includes the user's identity information, location coordinates information, time stamp, and other sensitive information. Although analyzing the users' locations and trajectories can better support MSNS and recommended services, it is easier for attackers to attack the user's location information so as to expose user's privacy. With the use of MSNS, mobile users are increasingly aware of the risk of privacy disclosure when enjoying location services [14–18].

Location-based privacy protection based on LBS is designed to prevent malicious attackers from gaining access

to the mobile user's location (or motion trajectory) to prevent user information from being compromised. Since the concept of LBS has been proposed, location-based privacy protection has quickly been raised and developed into academic research hot spots. The user's location privacy mainly includes location privacy, trajectory privacy, and user identity privacy. Aiming at these three aspects, quite a few privacy protection methods have been proposed, including commonly used dummy location technology, temporal and spatial anonymous technology, pseudonym technology and other methods protecting the users' location privacy, trajectory privacy, and identity information privacy [19–21]. Although the existing location privacy protection methods that apply to MSN can resist common privacy attacks [22], there are still some weaknesses to be resolved.

In this paper, we focus on the location privacy preserving in MSN aiming at larger communication overhead, larger range, and inaccuracy of query results for traditional anonymous schemes. The main contributions of this paper are summarized below.

(1) We propose a repartitioning anonymous region for location privacy preserving (RPAR) scheme based on the central anonymous location. The anonymous region is divided into several subregions, users' real locations are replaced by the central location, and a repartition is carried out to solve the tail anonymity user set after the anonymous region partition.

(2) We analyze the superiorities of RPAR algorithm. RPAR minimizes the communication traffic between the anonymous server and the LBS server while protecting the privacy of the user location. Furthermore, our scheme enables the users to obtain the more accurate query results, thus improving the quality of LBS.

(3) We simulate RPAR algorithm in extensive experiments. Simulation results show that our proposed scheme can effectively reduce the area of anonymous regions and minimize the traffic.

The rest of the paper is organized as follows. Section 2 reviews the related work. In Section 3, we give the preliminaries of location privacy preservation. Following that, in Section 4, the anonymous region repartition algorithm is proposed. In Section 5, simulations are given to verify the effectiveness of our proposed models and algorithms. Finally, we draw our conclusions and give the future work in Section 6.

2. Related Work

Recently, there have been considerable interests in the research of privacy preservation technologies for location-based services (LBSs) [23].

The k-anonymity scheme was first proposed to be Sweeney's application in a relational database [24] to protect sensitive properties by generalizing some of the nonsensitive properties in the database in case of stealing by adversaries. The generalization process makes the correlation between user information and their related record fuzzy, and every record in data table published is indistinguishable from other $k-1$ records. The scheme hides the link between the user and its corresponding information, guaranteeing the user's privacy information security.

To protect location privacy, researchers like Gruteser first proposed to apply the k-anonymity model to the location privacy preservation [25]. By constructing a location region composed of a query user and $k-1$ common users, the k users' geographical position information is generalized to be a k-anonymity region. The users being generalized cannot be distinguished from other $k-1$ users in terms of identity information, geographical position information, and so on. Even though the adversaries can access a user's location information in the anonymous region, they cannot identify the correspondence between the locations of the query information issued and the relevant user, protecting users' privacy information. The location anonymity scheme, proposed in [26], splits a contiguous anonymous region into several scattered subanonymity regions and demonstrates its effectiveness.

The privacy preserving scheme covers not only k-anonymity technology with strong applicability but also location privacy preserving scheme based on fake information. Bamba adopts virtual location information to hide real information in order to achieve location privacy preserving effect in [27]. Its main idea is as follows: Firstly users put forward location service requests and privacy requirements; then central anonymous servers send their fake position information and the real one to LBS servers; at last the LBS servers return query results back to anonymous servers, and the anonymous servers calculate the correct results and return them back to the corresponding users. Due to mixture of true and fake information, it is difficult for adversaries to distinguish the users' actual query information; thus the users' privacy information is protected.

Duckham et al. proposed to use obfuscation technique as an effective location privacy preserving mechanism [28] as well as position fuzzy region to process fuzzy query, satisfying the needs of privacy protection while sacrificing the quality of the service. Hence, how to strike a balance between location privacy preservation and service queries deserves our further study.

Ni et al. studied location privacy neighbor query based privacy preference in [29]; both the contradiction between the location protection and personal privacy preference and the balance between location privacy protection and query service quality problems are analyzed and summarized. Xu et al. proposed the location privacy region generation algorithm, centroid migration method [30], based on the spatial confusion location privacy protection technology; thus a certain deviation of centroid in the location privacy region was made. Ye et al. put forward a kind of location privacy protection scheme [31] based on active sharing mechanism; the main idea of this method is actively sharing the user's location information by sharing mechanism, so as to reduce the users' dependency on LBS servers and enhance the protective effect of the location privacy information.

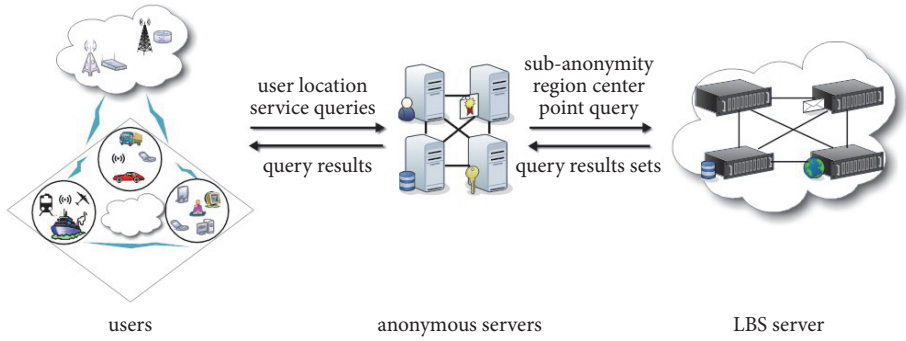


Figure 1: Architecture of our system model.

In recent years, domestic and foreign researchers and institutions have widespread attention on location-based privacy preservation; increasingly in-depth studies are carried out in location-based privacy. Besides the above-described location privacy preserving technologies, there are a wealth of methods such as location data randomization [32], fuzzification of space or time data [33, 34], methods based on strategies and encryption [35–37], and sensitive semantic based security anonymity mechanism [33, 38–40].

In [41], we have proposed a preliminary location privacy preserving scheme via repartitioning anonymous region in MSN. However, this scheme did not elaborate the design motivation and the algorithm analysis. Also, this scheme did not implement the simulations to verify its performance.

3. Preliminaries

The k-anonymity privacy preserving scheme was first applied to data releasing; extensive research and application are carried out after the application on the fields of location-based privacy preservation. Thereafter, anonymous region segmentation method is proposed to solve the insufficiency on large communication overhead of k-anonymity. However, in many cases, anonymous regions that are partitioned once cannot achieve ideal effect; therefore we put forward the repartitioning anonymous region scheme for location privacy preserving (RPAR).

3.1. System Model. At present, although there are many communication modes that are unsuitable for the central anonymous servers, current location privacy preserving schemes mainly adopt the central server mode [42, 43]. Our scheme uses the classic central server mode, and the anonymous queries are processed through both the central anonymous servers and the LBS servers. As shown in Figure 1, the principles of the model are as follows:

(1) When the user requests the location query service, all the query contents, location information, and parameters needed to be set are sent to the central anonymous servers.

(2) After receiving the query information sent by users, according to certain rules, the central anonymous servers will generate an anonymous user set which meets the requirements, figure out the number of subanonymity regions, and then partition them; a few scattered subanonymity regions are yielded. When the subanonymity regions meet the requirements, their central location is computed to replace corresponding subanonymity regions to send requests to the LBS server.

(3) The LBS server handles the query information sent by the central anonymous servers and returns the query results.

(4) After the refinement process, the central anonymous servers return the corresponding results to the users.

3.1.1. Relevant Definitions. To describe RPAR scheme, it is necessary to introduce some relative knowledge of location anonymity in the division process. According to the requirements of the scheme, we refer to the relevant definitions like location k-anonymity, nearest neighbor principles, etc., and several definitions such as central anonymity region location and the tail anonymity user set are also proposed.

Definition 1 ((location k-anonymity) [44]). Suppose that there exists a mobile user whose location coordinates is (x, y). If the user and at least other $k-1$ users cannot be differentiated by location information after the generalization for this user, we can say that the k users' locations satisfy location k-anonymity.

The k users' information forms a user anonymity set. Note that the least rectangular region which includes all the location K-anonymity users is called the location k-anonymity region. It is not difficult to see in Figure 2 that the rectangular box is an anonymous region with anonymous user set where $k = 12$.

Formally, we use AR to represent a location k-anonymity region. Thus, according to certain rules, AR can be divided into n discrete rectangular anonymity regions, represented by $AR = \{sub_{AR_1}, sub_{AR_2}, \ldots, sub_{AR_n}\}$. It should be clear that these small scattered anonymity regions $sub_{AR_i} (i = 1, 2, \ldots, n)$ are all the subanonymity regions of continuous anonymity region AR. Figures 2 and 3 depict the status before and after the anonymous regions are repartitioned, respectively.

Figure 2: Before partition of k-anonymity region.

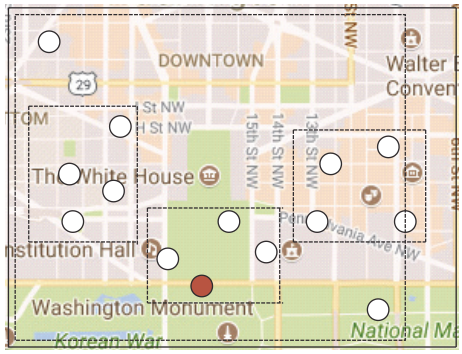


Figure 3: After partition of k-anonymity region.

Definition 2 (central location of anonymous region). The location of the two diagonals of a rectangular subanonymity region is said to be its central location, which is represented by coordinates (x', y').

We will take central location as a fake location to issue location service requests by replacing the subanonymity regions.

Definition 3 (tail anonymity user set). After partitioning the location k-anonymity region AR, if k/n is not an integer, then the supernumerary $k \mod n$ users form a tail anonymity user set.

Definition 4 (nearest neighbor principle). Take a location point as the center, and find other location points in accordance with the priority principle of the nearest Euclidean distance from the center point.

4. Location Repartitioning Anonymous Region Scheme

In order to overcome the drawbacks of the exiting schemes, including the oversized subanonymity regions and the inaccuracy of query results during the anonymous region partition, we propose the repartitioning anonymous region for location privacy preserving (RPAR) scheme. The oversized subanonymity regions are further partitioned, and the central locations of the subanonymity regions replace those subanonymity regions to issue queries to the servers, so as to reduce the communication overhead and achieve relatively accurate query results.

4.1. Motivation. According to the traditional anonymity schemes, when the users request LBSs, their true locations will be replaced with anonymous region and issue requests to LBS servers. However, if the anonymous parameter k is relatively bigger, then the anonymous regions will be relatively larger accordingly, especially apparent in the scenario where the users are sparse. Although the privacy preserving is in a high level, the accuracy of the request results is low, resulting in poor service quality.

Roman et al. [45] proposed anonymous partition algorithm for the first time, splitting the continuous k-anonymity region into several discrete subanonymity regions. Compared with the traditional continuous anonymity regions, the method in [45] narrows the region of the anonymous regions, improving the service quality to a certain extent. However, the final returned candidate result set is still fuzzy by querying through subanonymity regions, and the communication traffic has a certain decrease but the overhead is still large.

In addition, when there are still remaining users after the subanonymity regions are partitioned (i.e., there remains tail anonymity user set), the method in [45] puts all the remaining users in one subanonymity region. As shown in Figure 2, in the anonymous region where $k = 14$, the red solid dots represent the users who issue the query requests, and the white dots represent other real users in the k-anonymity region. We set the segmentation parameter $n = 3$, indicating that the anonymous user set is divided into three subanonymity regions, and the number of users in each subanonymity region is $k' = k/n = 4$, as shown in Figure 3. However, the following problems will occur when the scenario in subanonymity regions is consistent with that in Figure 3:

(1) The real locations of the users are replaced by the decentralized subanonymity regions in the corresponding regions to issued queries, which can still produce a large candidate result set and require a large amount of communication overhead.

(2) When the tail anonymity user set, that is, the value of $k \mod n$, is too large, it will result in the number of users in the subanonymity regions divided by the tail anonymous user set being larger than that in other subanonymity regions, resulting in great difference in query accuracy.

(3) As shown in Figure 3, if one of the subregions is oversized or even close to the area of the k-anonymity regions, the total area of subanonymity regions is greater than A_{min}, segmentation algorithm will return back to the k-anonymity algorithm with fewer users. The locations of the real users are far from the central region, causing queries to be not accurate.

In order to overcome these weak points, we construct RPAR scheme. Using the central location of the subanonymity regions instead of the subanonymity regions to issue queries to the servers, the communication traffic is

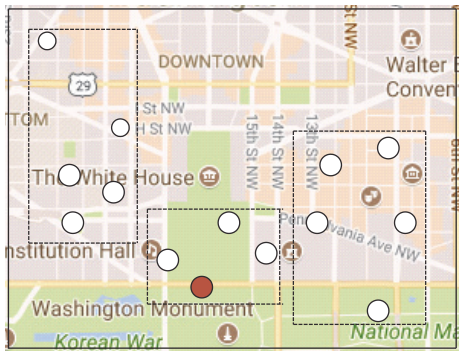


Figure 4: Repartition of anonymous region.

greatly reduced. When total area of subanonymity regions is greater than A_{min}, the largest subanonymity region is repartitioned into the other subanonymity regions, as shown in Figure 4, until the total area of the subanonymity regions is not greater than A_{min} anymore.

For the query accuracy problem caused by the oversized granularity of the tail anonymity user set, the users in the tail anonymity user set are partitioned into other subanonymity regions in accordance with the nearest neighbor principle.

4.2. Basic Idea of RPAR. Combined with Figures 2, 3, and 4, the basic idea of RPAR scheme is elaborated as follows:

(1) The solid red dots represent the users initiating query who find $k - 1$ users with the query users who can form k-anonymity regions according to the nearest neighbor principle, and k users information set is recorded. It can be seen in Figure 2 that $k = 14$.

(2) According to the parameter n, that is, the number of subanonymity regions, the k users are divided into n subanonymity regions, so the number of users each subanonymity region contains is $k' = k/n = 4$. The mobile users (red dots) are as the center to search other nearest $k' - 1$ users and form the first subanonymity region.

(3) A user from the rest of the users that is not in the first subanonymity region is randomly selected as the central point. According to the nearest neighbor principle, the subanonymity regions are formed with the user and other 3 users who are not in the first subregion from anonymous region, until the remaining user number is 0 or below k'.

(4) The tail anonymity user set is repartitioned into other subanonymity regions according to the nearest neighbor principle, and the subanonymity regions are updated.

(5) The area of the subanonymity regions is calculated. If the total area of the subanonymity regions is greater than A_{min}, users' maximum subanonymity region will be repartitioned until the total area of the subanonymity regions is not greater than A_{min}.

(6) The central locations of all the subanonymity regions are computed which are used to replace their subanonymity regions to issue queries to the LBS servers.

4.3. Algorithm Design. Let $Q = (id, L, S, M)$ be a user's location service query. For each Q, id represents the user identity, M represents the query information, L represents the location information, such as the user's location coordinates (x, y), and S is the privacy parameters including anonymous region parameter k, homogeneity parameter l (difference degree in anonymous query target regions), subanonymity region number n, and user's smallest total subanonymity area A_{min}. The parameters like k, l, and n are all user-defined. In practice, users can determine the values of of these parameters according to their own requirements and sensitivity. Thus, the algorithm can generate different degrees of anonymity to guarantee the performance.

For minimum area of anonymous regions, calculation probability is inferred through the Bayesian networks and the related background knowledge based on social network model, and minimum anonymous area is obtained by means of the maximum likelihood estimator. Figure 5 is the Bayesian network diagram for anonymous region estimation.

In the Bayesian network diagram, Q and k represent users' query information, A_{min} represents the minimum anonymous area, User is the user's query information, M is the user's query content such as hotel, hospital, *uid* and *position*, respectively, represent the user's identity and location information, *time* represents the query time, and l is the diversity parameter of query content which is not discussed in our paper.

In the process of calculation and estimation, without loss of generality, the relatively weaker nodes are ignored in order to simplify the computation complexity. The reasoning process of node A_{min} is described as follows:

(1) The probability distribution of the second and third nodes in the second layer is calculated:

$$P(User \mid e) \propto P\left(e_{c_j} \mid User\right) P(User \mid k) \propto \prod_{j=1}^{3} P\left(e_{c_j} \mid User\right) \quad (1)$$

$$P(User \mid e) = P\left(\frac{l}{M}\right) P(M \mid k) \propto P\left(\frac{l}{k} \mid M\right) \quad (2)$$

(2) Through the probability distribution of the first node in the first layer,

$$P(k \mid e) = P\left(e_{c_j} \mid k\right) = \prod_{j=1}^{3} P\left(e_{c_j} \mid k\right) \quad (3)$$

By (1) to (3), we have

$$P\left(A_{min} \mid k\right) = \left(\frac{P(k)}{P(User \mid e)}\right) P(M \mid e) \quad (4)$$

where e represents all the background knowledge and e_{c_j} represents the background knowledge of the child nodes.

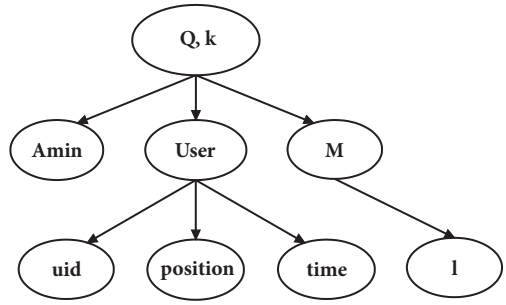


Figure 5: The Bayesian network for anonymous region estimation.

Input: uid, (x, y), k, l, n, A_{min}, M, other users' location information in anonymous set stored in anonymous servers.
Output: central locations (x_i, y_i) and M.

(1) $k' = \lfloor k/n \rfloor$
(2) form the first sub-anonymity region sub_{AR_1}, and calculate its *central location* (x_1, y_1) and area S_1 based on (x, y), k' and other user location information.
(3) **for** $i = 2$ **to** n **do**
(4) $m = k - k' * (i - 1)$;
(5) select user u_i randomly from the rest of the *tail anonymity user set* U_m: $u_i = Random(U_m)$;
(6) form sub_{AR_i} and and calculate its *central location* (x_i, y_i) and area S_i according to the information of u_i and parameter settings;
(7) calculate total area of all the sub-anonymity regions: $S_A = S_A + S_i$;
(8) **end for**
(9) **if** $k \mod n == 0$ **then**
(10) **if** $S_A > A_{min}$**then**
(11) repartition the largest sub-anonymity region until $S_A < A_{min}$
(12) **end if**
(13) **else**
(14) repartition the users in the *tail anonymous user set* into sub-anonymity regions according to the *nearest neighbor principle*
(15) **end if**
(16) update the *central locations* (x_i, y_i) of the sub-anonymity regions
(17) return (x_i, y_i) to all users in i-th sub-anonymity region. (x_i, y_i) and M are sent to the LBS servers to query location services.

Algorithm 1: RPAR.

Here, the background knowledge refers to maps, historical query records, and historical query probability of a certain area, etc.

When users query location service Q, all the parameters are needed to be sent to the anonymous servers. The parameters include identity information uid, location coordinates (x, y), anonymous region parameter k, subanonymity region number n, minimum anonymity regions' area A_{min}, query content M, and other users' location coordinates. After the parameters are processed by the anonymous servers, the fake locations are sent to the LBS servers to request service queries. In practice, the area of the anonymous region is calculated based on the coordinates of the position points of the user in the upper left and lower right corners of the anonymous region.

According to the algorithm description, parameter settings, and the algorithm process, the pseducode of the detailed RPAR algorithm is elaborated in Algorithm 1.

4.4. Algorithm Explication. RPAR is proposed based on the existing k-anonymity algorithm and partitioned subanonymity regions to solve the problems of the scattered subanonymity regions with total oversized area of anonymous regions and the oversized value of $k \mod n$.

In the process of the subanonymity region partition, the nearest neighbor principle is used which means searching qualified users according to the distance with nearest and highest priority. The distance in the nearest neighbor principle is Euclidean distance, namely,

$$d = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2} \tag{5}$$

where (a_1, b_1) and (a_2, b_2) are location coordinates of two points and d is the distance between two points.

According to the locations and parameters of the users who issued the location service requests, the servers find out the other $k - 1$ users nearest to this user, who satisfied the conditions. It is noted that, in the case of the repartitioning

of the subanonymity regions, the nearest neighbor principle is still used to make the request users as the center and form the first subanonymous region by searching the $k_1 - 1$ users, where k_1 is the user number in the first subanonymity region. Then the coordinates (x_1, y_1) of the central position in the first subanonymity region are calculated according ro the following formula:

$$x_1 = \frac{a_1 + a_2 + \cdots + a_k}{k_1}$$
$$y_1 = \frac{b_1 + b_2 + \cdots + b_k}{k_1} \quad (6)$$

where $(a_i, b_i), i = 1, 2, \ldots, k_1$, are each user's location coordinates in the first subanonymity region and k_1 is the users' number contained in this region.

After repartitioning the first subanonymity region, a user's location is selected randomly as the center. Other subanonymity regions are formed in accordance with the nearest neighbor principle, and the area and the coordinates of central locations of subanonymity regions are calculated. When n subanonymity regions are divided up, if $k \mod n = 0$, directly compare whether the total region of subanonymity regions is greater than A_{min}. If the answer is exact, then repartition the biggest anonymous region until the total region is not larger than A_{min}. If $k \mod n \neq 0$, then the users in the tail anonymity user set are repartitioned into the other subanonymity regions. After all the subanonymity regions are partitioned, the subanonymity regions are replaced with their central location to issue the location service requests to the LBS servers.

4.5. Algorithm Superiority Analysis. The traditional k-anonymity scheme makes k users form an anonymous region to issue queries to the LBS servers. The result of the queries generally is a candidate set. In the scenario with sparse users, the area of the anonymous regions will be large and the searched candidate set will also be correspondingly larger. Li et al. [26] divided the anonymous region into a few scattered subanonymity regions, using each subanonymity region to query; the query candidate set is small accompanied with precision improved; however a lot of communication overhead would cost.

Our RPAR scheme directly splits k users into several distributed subanonymity regions instead of forming a k-anonymity region. The regions are replaced with its center to issue requests to the LBS servers, which greatly reduce the communication overhead, and the accurate query results are obtained which referred to the central location of subanonymity regions, so each user can get their accurate query results by referring to the location. Therefore, compared with the traditional methods, the advantages of RPAR are obvious, especially when the users are sparse.

After dividing the anonymous regions [26], if $k \mod n \neq 0$, then the rest of the users are put into one of the anonymous regions. RPAR scheme repartitions the rest of the users into their nearest subanonymity regions in accordance with the nearest neighbor principle.

When the value of $k \mod n$ is relatively small, no matter how to divide the remaining users, the number of users in each anonymous region will not make a big difference, so the query precision degree of results returned also will not make a big difference.

While when the value of $k \mod n$ is larger, the method in [26] will lead to one anonymous region far larger than the others, the query precision has great difference. In our RPAR, after $k \mod n$ users are repartitioned according to the nearest neighbor principle, the users in each subanonymity region are much the same, so the result precision degree of the final queries will not be too different.

In this paper, the condition that the total area of subanonymity region is greater than A_{min} is also considered in subanonymity region partition. The value of A_{min} is defined by the user, which generally is the area of the k-anonymity regions. As shown in Figure 3, if some of the area of subanonymity regions is too large or even close to that of the K-anonymity regions, the partition algorithm will be returned back to the k-anonymity algorithm with fewer users. Therefore, when the total area of the subanonymity regions is greater than A_{min}, the biggest subanonymity region is partitioned until the total area of the subanonymity regions is less than A_{min}.

We use the central locations of the subanonymity regions instead of subanonymity regions to issue query requests to the LBS servers. As fake positions, the central locations well hide the true users' locations. Considering the condition that some central positions would coincide with users' actual locations, due to the scheme that all users in the subanonymity regions use fake positions, even if the query results are intercepted by adversaries, identification probability of actual users is at least $1/k_i$.

5. Simulations and Performance Analysis

In this section, we implement RASA and analyze its performance.

5.1. Simulation Environment. As for the experimental environment, Windows 7 with 64-bit operating system with 4G RAM and Intel (R) Pentium (R) CPU G2030@3.00 GHz processor is adopted. Thomas Brinkhoff's mobile object generator [46] is used to generate the spatiotemporal data as the resulting dataset. The privacy preserving parameters of the experiment are determined according to the users' requirements.

5.2. Simulations and Analysis. Assume that the mobile users are deployed in a rectangular region with an area of $5000m * 5000m$. Since the effect of our RPSA scheme is obvious for the scenario with sparse users, the simulation results are generated in this scenario with our RPSA scheme.

In the following, we will compare our RPAR algorithm with the traditional k-anonymity, FCR [26], and DSCR [47] algorithms. FCR directly generates the total area of the subanonymity regions during the partition while the remaining $k \mod n$ users are put in the same subanonymity region. However, DSCR firstly generates contiguous anonymous

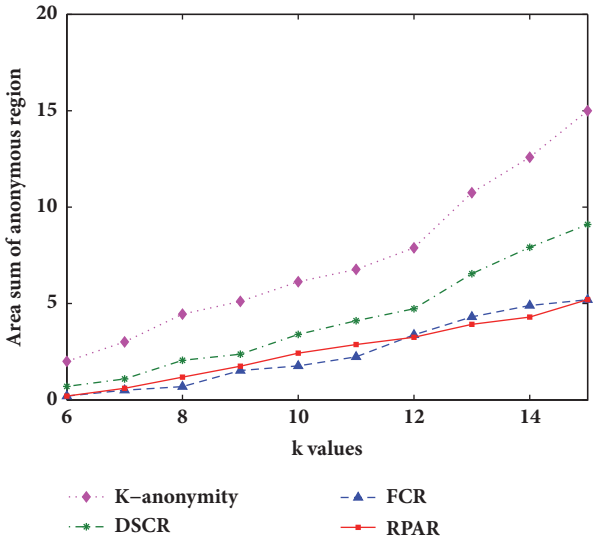


Figure 6: Area comparison of anonymous regions.

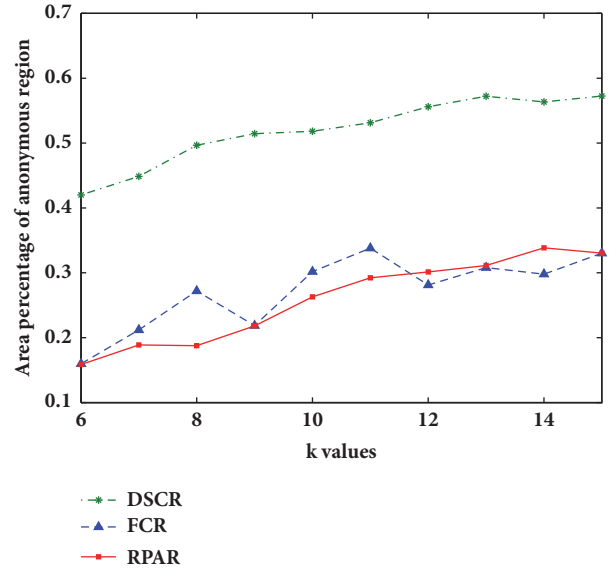


Figure 7: Region percentage comparison of anonymous regions.

regions; then continuous anonymous regions are divided into several subanonymity regions which may show a large number of repetitive regions.

We analyze the performance of RPAR by comparing the anonymous area, anonymous area percentage, and anonymous time of these methods with the same anonymous parameters. We assume that the value of the parameter k in the simulation is an integer in the range from 6 to 15. We also set the number of subanonymity regions n as 3. The comparison diagrams are illustrated in Figures 6, 7, and 8.

As shown in Figure 6, the area in the anonymous regions produced by our RPAR is not much different from that in FCR, whereas the area is greatly reduced compared with traditional k-anonymity and DSCR. While $k \mod n \neq 0$ in FCR, FCR puts the rest of the $k \mod n$ users into a subanonymity region; thus this area is oversized, causing the user query precision of this anonymous region and other subanonymity regions to vary greatly. Fortunately, our RPAR effectively solves the above problem. Note that, with the k value increasing, the area increases more slowly and the reduced area is increasingly large in RPAR compared with other three anonymous region partition schemes, making RPAR superior obviously.

Figure 7 is the comparison of the ratio of the effective area of anonymous region generated by DSCR, FCR, and RPAR with the traditional k-anonymity, namely,

$$per_2 = \frac{Area_{DSCR}}{Area_{k-anonymity}},$$
$$per_3 = \frac{Area_{FCR}}{Area_{k-anonymity}}, \quad (7)$$
$$per_4 = \frac{Area_{RPAR}}{Area_{k-anonymity}}.$$

It can be seen from Figure 7 that the area ratio of DSCR is larger, and the area ratio of FCR is closer to that of RPAR.

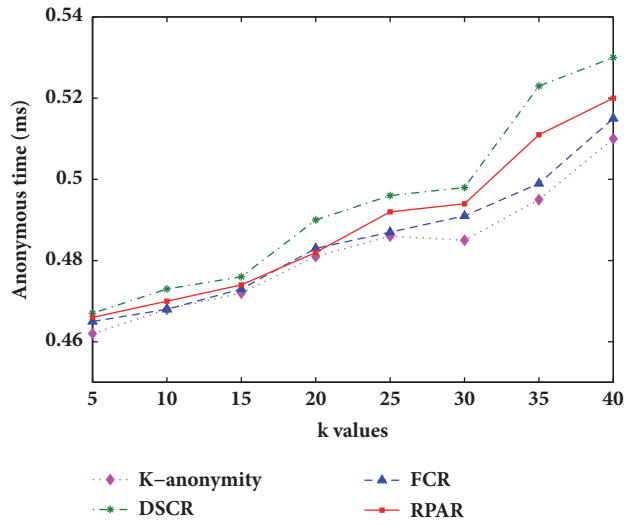


Figure 8: Comparison of anonymous time.

As a whole, the size of k is positively related to per_i, where $i = 2, 3, 4$. With the increase of k, the area of valid anonymous regions becomes increasingly large and stable. However, our RPAR scheme performs better than FCR as the value of $k \mod n$ is larger gradually. We observe that RPAR can also obtain stable and higher quality service while reducing the area of anonymous regions, whereas FCR scheme is not stable.

Figure 8 compares the time of anonymous regions in the algorithms. Traditional k-anonymity scheme directly generates consecutive k-anonymity regions; DSCR firstly generates k-anonymity regions which are then partitioned into subanonymity regions, and the $k \mod n$ tail-anonymity regions are processed; therefore, the anonymous time is longer than the traditional k-anonymity. More specifically, anonymous time is relatively short when $k \mod n = 0$ while

the anonymous time is relatively long when $k \mod n \neq 0$. FCR directly generates subanonymity regions, which results in the anonymous time being shorter. Note that our RPAR needs to deal with the tail anonymous user set in the process of directly generating subanonymity regions; therefore the time may be longer than both traditional k-anonymity and FCR, but less than DSCR. Fortunately, the additional average time is less than 0.1 *ms*, which is quite small, and it will not make a big difference to the algorithm performance.

6. Conclusions and Future Work

Aiming at large communication overhead, large range, and inaccuracy of query results for traditional anonymous schemes, this paper proposed an anonymous region repartition algorithm by studying the users' location privacy preservation. The anonymous region is divided into several subregions, the users' real locations are replaced by the central location, and a repartition is carried out to solve the remaining users' region set after the anonymous region segmentation. Finally, the algorithm is analyzed and the privacy degree is evaluated, and the simulation experiment is carried out. Experimental results show that the proposed scheme has some advantages in privacy anonymity.

In the future, we will research the location privacy preserving in the scenario of dense region. Furthermore, we will research the location privacy preserving under distributed environment.

Data Availability

Thomas Brinkhoff's mobile object generator is used to generate the spatiotemporal data as the resulting dataset in our paper, and the website is http://iapg.jade-hs.de/personen/brinkhoff/generator/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National Key R&D Programs Project of China under Grant 2017YFC0804406, NSF of China under Grant 61672321, 61771289, and 61373027, Training Program of the Major Research Plan of NSF of China under Grant 91746104, Project of Shandong Province Higher Educational Science and Technology Program under Grant J15LN19, and Open Project of Tongji University Embedded System and Service Computing of Ministry of Education of China under Grant ESSCKF 2015-02.

References

[1] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng, "A Privacy Preserving Communication Protocol for IoT Applications in Smart Homes," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1844–1852, 2017.

[2] R. Li, T. Song, N. Capurso, J. Yu, J. Couture, and X. Cheng, "IoT Applications on Secure Smart Shopping System," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1945–1954, 2017.

[3] T. Song, N. Capurso, X. Cheng, J. Yu, B. Chen, and W. Zhao, "Enhancing GPS with lane-level navigation to facilitate highway driving," *IEEE Transactions on Vehicular Technology*, vol. 99, 2017.

[4] C. Hu, H. Li, Y. Huo, T. Xiang, and X. Liao, "Secure and Efficient Data Communication Protocol for Wireless Body Area Networks," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 2, pp. 94–107, 2016.

[5] C. Hu, N. Zhang, H. Li, X. Cheng, and X. Liao, "Body area network security: a fuzzy attribute-based signcryption scheme," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 37–46, 2013.

[6] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, and D. Chen, "OPFKA: secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks," in *Proceedings of the INFOCOM, 2013 Proceedings IEEE*, pp. 2322–2330, 2013.

[7] 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*, 2016.

[8] C. Hu, W. Li, X. Cheng, J. Yu, S. Wang, and R. Bie, "A Secure and Verifiable Access Control Scheme for Big Data Storage in Clouds," *IEEE Transactions on Big Data*, pp. 1-1, 2017.

[9] X. Zheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Follow but no track: privacy preserved profile publishing in cyber-physical social systems," *IEEE Internet of Things Journal*, 2017.

[10] A. Alrawais, A. Alhothaily, J. Yu, C. Hu, and X. Cheng, "SecureGuard: A Certificate Validation System in Public Key Infrastructure," *IEEE Transactions on Vehicular Technology*, 2018.

[11] X. Zheng, Z. Cai, G. Luo, L. Tian, and X. Bai, "Privacy-preserved community discovery in online social networks," *Future Generation Computer Systems*, 2018.

[12] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[13] G. Sun, Y. Xie, D. Liao, H. Yu, and V. Chang, "User-defined privacy location-sharing system in mobile online social networks," *Journal of Network and Computer Applications*, vol. 86, pp. 34–45, 2017.

[14] X. Xiao, C. Chen, A. K. Sangaiah, G. Hu, R. Ye, and Y. Jiang, "CenLocShare: A centralized privacy-preserving location-sharing system for mobile online social networks," *Future Generation Computer Systems*, vol. 86, pp. 863–872, 2018.

[15] H. Shen, G. Bai, M. Yang, and Z. Wang, "Protecting trajectory privacy: A user-centric analysis," *Journal of Network and Computer Applications*, vol. 82, pp. 128–139, 2017.

[16] X. Zheng, G. Luo, and Z. Cai, "A fair mechanism for private data publication in online social networks," *IEEE Transactions on Network Science and Engineering*, 2018.

[17] H. Chen and W. Lou, "On protecting end-to-end location privacy against local eavesdropper in wireless sensor networks," *Pervasive and Mobile Computing*, vol. 16, pp. 36–50, 2015.

[18] Y. Liang, Z. Cai, Q. Han, and Y. Li, "Location privacy leakage through sensory data," *Security and Communication Networks*, vol. 2017, Article ID 7576307, 12 pages, 2017.

[19] M. L. Damiani, "Privacy enhancing techniques for the protection of mobility patterns in LBS: Research issues and trends," in *Security and Communication Networks*, vol. 244, pp. 223–239, Springer Netherlands, 2012, European Data Protection: Coming of Age.

[20] K. Xing, C. Hu, J. Yu, X. Cheng, and F. Zhang, "Mutual privacy preserving *k*-means clustering in social participatory sensing," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2066–2076, 2017.

[21] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, "Cost-efficient strategies for restraining rumor spreading in mobile social networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2789–2800, 2017.

[22] X. Liang, K. Zhang, X. Shen, and X. Lin, "Security and privacy in mobile social networks: challenges and solutions," *IEEE Wireless Communications Magazine*, vol. 21, no. 1, pp. 33–41, 2014.

[23] A. Sadilek, H. Kautz, and J. P. Bigham, "Finding your friends and following them to where you are," in *Proceedings of the 5th ACM International Conference on Web Search and Data Mining, WSDM 2012*, pp. 723–732, USA, February 2012.

[24] L. Sweeney, "*k*-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[25] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pp. 31–42, ACM, San Francisco, Calif, USA, 2003.

[26] T. C. Li and W. T. Zhu, "Protecting user anonymity in location-based services with fragmented cloaking region," in *Proceedings of the EEE Conf. on Computer Science and Automation Engineering (CSAE)*, vol. 3, pp. 227–231, Zhangjiajie, China, 2012.

[27] B. Bamba, L. Liu, P. Pesti et al., "Supporting anonymous location queries in mobile environments with privacy grid," in *Proceedings of the 17th ACM Inter. Conf. on World Wide Web (WWW)*, pp. 237–246, Beijing, China, 2008.

[28] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Proceedings of the Inter. Conf. on Pervasive Computing (PC)*, pp. 152–170, Munich, Germany, 2005.

[29] W. Ni and X. Chen, "Research on privacy preference in privacy location," *Journal of Software*, vol. 27, no. 7, pp. 125–130, 2016.

[30] H. Xu, J. Xu, Y. Gong et al., "Location privacy region generation algorithm based on spatial confusion location privacy protection," *Journal of South China Univer. of Technol.: Natural Science Edition*, vol. 42, no. 1, pp. 97–103, 2014.

[31] A. Ye, S. Lin, J. Ma et al., "An active diffusion location privacy protection method," *Acta Electronica Sinica*, vol. 43, no. 7, pp. 1362–1368, 2015.

[32] R. Kato, M. Iwata, T. Hara et al., "A dummy-based anonymization method based on user trajectory with pauses," in *Proceedings of the 20th ACM Inter. Conf. on Advances in Geographic Information Systems (AGIS)*, pp. 249–258, California, USA, 2012.

[33] E. Yigitoglu, M. L. Damiani, O. Abul, and C. Silvestri, "Privacy-preserving sharing of sensitive semantic locations under road-network constraints," in *Proceedings of the 13th IEEE Inter. Conf. on Mobile Data Management (MDM)*, pp. 186–195, 2012.

[34] X. Pan, J. L. Xu, and X. F. Meng, "Protecting location privacy against location-dependent attacks in mobile services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1506–1519, 2012.

[35] D. Chen, P. Zhang, and C. Hu, "Private and precise range search for location based services," in *Proceedings of the IEEE Inter. Conf. on Commun. (ICC)*, pp. 7347–7352, China, 2015.

[36] L. Li, R. Lu, and C. Huang, "EPLQ: Efficient Privacy-Preserving Location-Based Query over Outsourced Encrypted Data," *IEEE Internet of Things Journal*, vol. 3, no. 2, pp. 206–218, 2016.

[37] I.-T. Lien, Y.-H. Lin, J.-R. Shieh, and J.-L. Wu, "A novel privacy preserving location-based service protocol with secret circular shift for k-NN search," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 863–873, 2013.

[38] M. Li, Z. Qin, and C. Wang, "Sensitive semantics-aware personality cloaking on road-network environment," *International Journal of Security and Its Applications*, vol. 8, no. 1, pp. 133–146, 2014.

[39] C. Hu, A. Alhothaily, A. Alrawais, X. Cheng, C. Sturtivant, and H. Liu, "A secure and verifiable outsourcing scheme for matrix inverse computation," in *Proceedings of the IEEE Conference on Computer Communications(CC)INFOCOM*, 2017.

[40] 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*, 2018.

[41] L. Ni, Y. Yuan, X. Wang, M. Zhang, and J. Zhang, "A Location Privacy Preserving Scheme Based on Repartitioning Anonymous Region in Mobile Social Network," *Procedia Computer Science*, vol. 129, pp. 368–371, 2018.

[42] L. Zhang, Z. Cai, and X. Wang, "Fakemask: A novel privacy preserving approach for smartphones," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 335–348, 2016.

[43] R.-H. Hwang, Y.-L. Hsueh, J.-J. Wu, and F.-H. Huang, "SocialHide: A generic distributed framework for location privacy protection," *Journal of Network and Computer Applications*, vol. 76, pp. 87–100, 2016.

[44] D. Yang, X. Fang, and G. Xue, "Truthful incentive mechanisms for k-anonymity location privacy," in *Proceedings of the 32nd IEEE Int. Confer. on Comput. Commun.INFOCOM, Italy*, pp. 2994–3002, 2013.

[45] R. Yarovoy, F. Bonchi, L. V. Lakshmanan et al., "Anonymizing moving objects: How to hide a mob in a crowd?" in *Proceedings of the ACM 12th Inter. Conf. on Extending Database Technology (EDT): Advances in Database Technology*, pp. 72–83, Saint Petersburg, Russia, 2009.

[46] "Thomas Brinkhoff Network-based Generator of Moving Objects," http://iapg.jade-hs.de/personen/brinkhoff/generator/.

[47] Z. Zhao, L. Li, F. Zhang et al., "Location privacy protection method based on dispersive anonymous region," *Journal of Shandong Univ. (Natural Science)*, vol. 48, no. 7, pp. 56–61, 2013.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 4248092, 14 pages
https://doi.org/10.1155/2018/4248092

Research Article

Research on Trajectory Data Releasing Method via Differential Privacy Based on Spatial Partition

Qilong Han, Zuobin Xiong, and Kejia Zhang

Department of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Kejia Zhang; kejiazhang@hrbeu.edu.cn

Received 23 August 2018; Revised 30 September 2018; Accepted 9 October 2018; Published 1 November 2018

Guest Editor: Liran Ma

A number of security and privacy challenges of cyber system are arising due to the rapidly evolving scale and complexity of modern system and networks. The cyber system is a fundamental ingredient for Internet of Things (IoT) and smart city which are driven by huge amount of data. These data carry a lot of information for mining and analysis, especially trajectory data. If unprotected trajectory data is released, it may disclose user's personal privacy, such as home, religion, and behavior mode, which will endanger their personal security. Until now, many methods for protecting trajectory information have been proposed. However, these methods have the following deficiencies: (i) they cannot defend against speculative attacks if the attacker's background knowledge is maximized; (ii) when studying the problem, they made some strong assumptions that did not match the reality; (iii) the implementation algorithm is complicated and the time complexity is high, which means that data cannot be executed quickly when the amount is large. So, in this paper, we propose a spatial partition based method to publish trajectory data via differential privacy. First, by exponential mechanism, we divide location set at the same time into different partitions fast and accurately. Then we propose another effective method to release trajectory in a differential private manner. We design experiment based on the real-life dataset and compare it with existing method. The results show that the trajectory dataset released by our algorithm has better usability while ensuring privacy.

1. Introduction

In recent years, with the development of IoT, smart city is becoming popular to us and facilitates our life. As the important foundation of IoT application, cyber-physical system collects and provides a lot of data to it from users. Usually, personal data of user include the real-time location, usage time, and biometric information [1]. Among them, trajectory information is very important to user as well as adversary. Because it carries a lot of information for data mining and scientific analysis, if the trajectory is obtained illegally or released without protection, it is easy to reveal the user's personal privacy, such as home address and behavior mode, which will endanger the personal security of the user [2–4]. Once the information is leaked into an attacker, it will cause immeasurable loss to the user, resulting in threats to personal safety and property. Therefore, in cyber system, for the security and privacy of user, it is extremely urgent to provide an effective protection method for a large amount of users' privacy data [5–11].

Privacy disclosure and protection in trajectory data publication can usually be divided into two categories [12]:

(1) Only one trajectory is included in the published trajectory dataset. Each location point on the trajectory corresponds to a record, and the user's privacy requirement is to ensure that the location at a certain point is safe [13].

(2) The published trajectory dataset contains multiple trajectories. Each of these trajectories is considered a record. Our aim is to publish a sanitized dataset so that the attacker could not know the correspondence between a trajectory and the user.

Based on the above two types of problems, many privacy protection methods based on *k-anonymity* [14] and partitioning have emerged in recent years, such as *l-diversity* [15], *t-closeness* [16], and (α, k)- *anonymity*. Although these methods can protect more details of the data, they all require special attack assumptions and background knowledge. In addition, for the above privacy protection methods, some new attack models have emerged, such as *combined attacks* [17] and

background knowledge attacks [18]. These new attack models are really serious challenges to the effectiveness of above methods.

The root cause of above situation is that (i) the background knowledge of attacker is difficult to define and (ii) these early privacy protection models did not provide an effective and rigorous way to prove their level of privacy protection. Therefore, researchers are trying to find a sufficiently usable privacy protection model that can resist various forms of attack with the attacker's maximum background knowledge. The rise of differential privacy (DP) [19] makes it possible to implement this idea.

Differential privacy is a probability-based privacy model proposed by Dwork [19] in 2006 for privacy breach of statistical database. The advantages of differential privacy are the following: (i) it is based on a powerful mathematical model that can provide quantitative analysis for privacy level; (ii) the usability can be controlled by adjusting privacy budget to add proper noise; (iii) the privacy can still be guaranteed even if the attacker's background knowledge is maximum. Because of these above advantages, differential privacy has quickly gained the attention of researchers [20].

The main content of this paper is to use differential privacy to protect the trajectory dataset generated by moving objects. For the spatiotemporal trajectory dataset, it is important to know how to use the differential privacy method to process the data, so that the published data can protect the relationship between the user and the trajectory while protecting the sensitive location of the user on the trajectory.

Contributions. The main contributions of this paper are threefold:

(1) For original dataset in which location points are strictly ordered by timestamp, we propose a Hilbert curve based spatial partition method. According to the sparsity of location distribution in the area at the same time, we leverage exponential mechanism to get the most likely accurate partition, which protects sensitive locations of individual moving object.

(2) We then propose a simple and effective differential privacy data publishing algorithm to generate trajectory. On the basis of the partitions of each timestamp, we generate similar trajectories of original trajectory. Then we design a method to reduce the Laplace noise injected into the data. The released noisy dataset protects the relationship between the user and trajectory.

(3) Through theoretical analysis and experimental evaluation on the real-life dataset, the privacy guarantee and usability of the proposed publishing method are proven. Compared with the existing algorithm, using the Hausdorff distance and spatiotemporal range query distortion as the evaluation criteria, the experimental results show that our algorithm is superior to the previous algorithm.

2. Related Work

Researchers have conducted a lot of researches on trajectory privacy protection and achieved rich results. Nicolás [21] pointed out that directly deleting the ID of the trajectory does not guarantee the user's privacy. With the advancement of the attack means, even if the location of each user is protected, the attacker can still learn the user's mode through association analysis and data mining.

In such methods, trajectory *k-anonymity* is one of the most commonly used methods [1]. Sweeney first proposed the *k-anonymity* model [14] and at least k-1 records were indistinguishable from each record. This method guarantees user privacy to some extent, but the sensitive attribute values in the same anonymous set may be identical or of few types, and the attacker can still infer the information of a record from the table. In response to this problem, Shwin Bgl et al. [15] proposed a privacy protection standard for *l-diversity*, which requires that each k-anonymous set has at least l different sensitive attribute value under the premise that the data record satisfies the *k-anonymity* model. This prevents an attacker from matching a record to a determined individual. In [22, 23], the trajectory *k-anonymity* is extended, and the (k, δ)-*anonymity* model is proposed. It is required to find at least k-1 other trajectories in the δ uncertainty region around it. In [24], by suppressing some sensitive information in the user's trajectory, the probability of the attacker acquiring the user's trajectory information through data mining is reduced, so the trajectory privacy is protected. Terrovitis [25] proposed a new suppression mechanism, which divides the trajectory into sensitive regions and nonsensitive regions. When the user enters the sensitive area, the user's location in the area is suppressed, and the information update is stopped; when entering the nonsensitive area, no suppression is performed. Kido [26] proposed a trajectory data suppression mechanism that can simultaneously suppress sensitive nodes in the trajectory and nodes that can uniquely identify users to achieve trajectory privacy protection. Mohammed [27] designed a user trajectory suppression mechanism for high-dimensional sparseness and selected appropriate suppression points combined with *k-anonymity,* so that each subsequence of the same length in the user trajectory has no less than k identical subsequences. Shokri [28] proposes a way to add random noise to the user's actual location by comprehensively considering the user's privacy requirements, the attacker's knowledge, and the maximum tolerable quality of service degradation caused by the confusion of the real location. When attacker reconstructs the actual location of the user, the error rate of speculating rises while satisfying the user's quality of service requirements. According to the predictability of human behavior patterns, Theodorakopoulos [29] proposed an algorithm for exchanging access-sensitive location points to predict the user's future trajectory while protecting the privacy of the user's location.

After the differential privacy model is proposed, it was quickly applied in the privacy protection of trajectory data release. In [30], the differential privacy model was applied for the first time to propose the prefix method. This method uses the hierarchical framework to construct the prefix tree, dividing the trajectory sequence with the same prefix into the same branch of the tree and adding the noise by counting the node's count value. However, as the tree grows, the prefix will form a large number of leaf nodes, making the added noise too large and reducing the accuracy of the published

Table 1: Notations that will be used.

Notation	Explanation
T_i	a trajectory in dataset
$\widetilde{T}_i$	generalized trajectory in noisy dataset
D	dataset has multiple trajectories
r	radius of the stand deviation circle
r_w	radius of the weighted stand deviation circle
L, L_i	location universe and the location set at each timestamp i
$\widetilde{L}, \widetilde{L_i}$	location universe and location set at timestamp i after dividing
L_{ij}	center of location set after dividing in the timestamp i, jth set
C	set of trajectory count number in dataset
d	distance set in adjacent trajectory count number
$\widetilde{D}, \overline{D}$	the noisy dataset and the published dataset

dataset. Also, these above methods only consider the spatial characteristics of the trajectory data, regarding the trajectory data as a sequence of spatial location points, or make bad utility when trajectory is long.

3. Materials and Methods

In this chapter, we define the spatiotemporal trajectory dataset, review the knowledge of differential privacy, and introduce a few methods that we will use in the next part. Besides, notations we use are listed in Table 1.

3.1. Spatiotemporal Trajectory Database

Definition 1 (spatial-temporal trajectory). A spatiotemporal trajectory is a location sequence generated by ordering multiple timestamps, representing the trajectory of a moving object in space. $T = (l_1, t_1) \longrightarrow (l_2, t_2) \longrightarrow \cdots \longrightarrow (l_{|T|}, t_{|T|})$, where $|T|$ is the length of this trajectory, and, $\forall i$ $(1 \leq i \leq |T|)$, $l_i \in L_i$ is a discrete spatial point, which is represented by latitude and longitude coordinate.

L_i is the location universe of locations at time t_i. We use $Time(T)$ to represent the timestamps of a spatiotemporal trajectory.

A spatiotemporal trajectory dataset D is a dataset consisting of $|D|$ spatiotemporal trajectories, like $D = \{T_1, T_2, \cdots, T_{|D|}\}$. In general, the length of each trajectory and the sampling interval are different due to the different sources and sampling methods of the dataset. For the sake of convenience, the original dataset will be preprocessed in our experiments, making

$$Time\left(T_i\right) = Time\left(T_j\right), \quad \forall T_i, T_j \in D,\ i \neq j. \tag{1}$$

3.2. Differential Privacy. As a well-defined and provable privacy model, differential privacy has been widely used in data protection and data mining privacy protection since it was introduced [20].

Definition 2 (ε-differential privacy). A randomized algorithm is differential privacy if and only if any two databases D and D' contain at most one different record and, for any possible anonymized output $O \in Range(A)$,

$$\Pr\left(A\left(D\right) = O\right) \leq e^{\varepsilon} \times \Pr\left(A\left(D'\right) = O\right) \tag{2}$$

We say that the algorithm satisfies ε-differential privacy.

In the above formula, ε is a parameter. The smaller it is, the stronger the privacy protection provided by the differential privacy mechanism is.

In this paper, two important tools for implementing differential privacy are the Laplace mechanism [31] and the exponential mechanism [32]; both of them use the global sensitivity.

Definition 3 (global sensitivity). For a given function $f : D \longrightarrow R^d$, its global sensitivity is

$$\Delta f = \max_{D_1, D_2} \left\| f\left(D_1\right) - f\left(D_2\right) \right\|_1 \tag{3}$$

D_1 *and* D_2 are neighboring databases that differ in at most one record.

Laplace Mechanism. The mechanism is designed for functions whose query results are numerical. Differential privacy is achieved by injecting appropriate noise into the result of query. The noise generation is based on the Laplace distribution function and its probability distribution is

$$p\left(x \mid b\right) = \frac{1}{2b} e^{-|x|/b}, \quad b = \frac{\Delta f}{\varepsilon} \tag{4}$$

where b is the noise scale and it is determined by the global sensitivity of the function and the privacy budget.

Theorem 4. *For any $f : D \longrightarrow R^d$, the mechanism that adds independently generated by Laplace noise on the real result, like*

$$A\left(D\right) = f\left(D\right) + Laplace\left(b\right) \tag{5}$$

Then the mechanism satisfies ε-differential privacy.

Exponential Mechanism. For some functions whose query result is nonnumeric or has no meaning after adding noise,

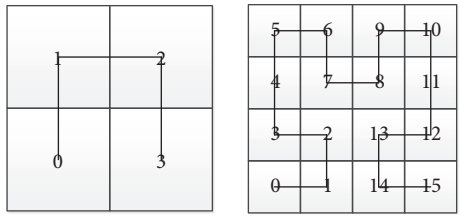


Figure 1: 1- and 2-order Hilbert curve.

such as the query result which is a certain attribute in the dataset, Mcsherry and Talwar [32] proposed a protection mechanism that satisfies differential privacy for this situation. It first defines utility function $u : (D \times r) \longrightarrow R$ that assigns a real score for every possible output r in the output domain R. Higher score means more utility. It then selects an output $r \in R$ with the probability proportional to $e^{(\varepsilon \times u(D,r))/2\Delta u}$; $\Delta u = \max_{\forall r, D_1, D_2} |u(D_1, r) - u(D_2, r)|$ is the sensitivity of utility function. As the outputs with higher score are more likely to be selected, this mechanism is close to optimal. In addition, the utility function should be insensitive to the change of a single record.

Theorem 5. *For any function $u : (D \times r) \longrightarrow R$, the mechanism chooses an output r, $r \in R$, with the probability proportional to $e^{(\varepsilon \times u(D,r))/2\Delta u}$ being able to guarantee ε-differential privacy.*

Composition Properties

Theorem 6 (sequential composition). *Suppose that each algorithm A_i satisfies ε_i-differential privacy. A sequence of A_i over database D provides $\sum \varepsilon_i$-differential privacy.*

Theorem 7 (parallel composition). *Suppose that each algorithm A_i satisfies ε_i-differential privacy. A sequence of A_i over a set of disjoint database D_i provides* $\max(\varepsilon_i)$*-differential privacy as a whole.*

3.3. Hilbert Curve. The Hilbert space filling curve is a continuous but nonconductible mathematical curve proposed by German mathematician Hilbert in 1891 [33], which is being widely applied in spatial sorting currently. Space filling curve is a method of mapping d-dimensional space into 1-dimensional space. It passes through every discrete unit of high-dimensional space only once and numbers these units in a linear order as in Figure 1. From the d-dimensional space to the linear space mapping process, the Hilbert curve can maintain location relationship between the point and its neighbors to some extent, so it has good characteristics in the spatial point division problem [34].

Using Hilbert curve to divide the space is relatively fast and capable of resisting speculative attacks [35]. However, due to the uneven distribution of spatial points and sparsity, the space of the Hilbert curve partition is slightly larger than the space based on the *KNN* method [36], which results in larger errors in the calculation of spatial point anonymity and relatively high computational time. Therefore, we need an efficient method that maintains both the spatial nature of the Hilbert curve and the ability to produce relatively small partition.

We note that points with similar Hilbert values are similar in the 2-dimensional or high-dimensional space is a sufficient unnecessary condition [37], which means that, in a Hilbert curve, there may be similar points in the two-dimensional space but the Hilbert values differ greatly. In Figure 2(a), U_2 and U_8 are such points. And because the Hilbert curve is recursively divided using the quadrant mode, this results in no Hilbert curve that makes the two points close, no matter how high order there is. This is why the space partition in Hilbert curve is slightly larger than the space based on the *KNN* method. We also noticed that, for spatial points under the same distribution, even the equal-order Hilbert curves could provide different partition after rotating. In Figure 2, the eight users are in the same locations, but the spatial regions obtained after 3-anonymous partitioning [38] through different 3-order Hilbert curves have a large difference in size.

In Figure 2(a), the partition of U_1, U_2, U_3 occupies 16 cells and U_4, U_5, U_6, U_7, U_8 have 49 cells. The average is 32.5. Meanwhile, in Figure 2(b), U_5, U_6, U_7 occupies 6 cells and U_8, U_2, U_3, U_4, U_1 20. The average is 13. The effect after the rotation is visible. Therefore, this paper considers using such method to divide the space. It is hoped that when the divided regions are constructed, the accuracy of the partitions can be ensured while obtaining a smaller divided region.

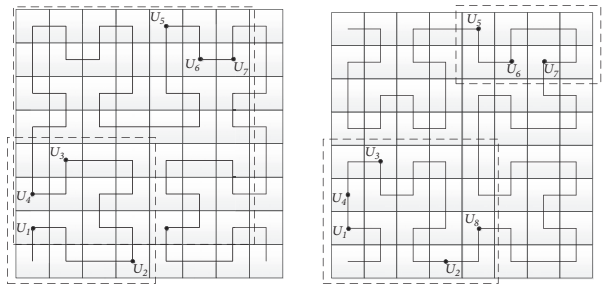
3.4. Location Entropy. The definition of location entropy is derived from Shannon entropy [39] and is a measure of uncertainty. In the field of location privacy, many methods [40, 41] have adopted the concept of location entropy to measure the popularity of points of interest (POI) or to use location entropy to design privacy protection mechanisms.

For a given location l, let V_l be the set of visits to that location. Thus, $c_l = |V_l|$ is the total number of visits to l. Also, let U_l be the set of distinct users that visited l, and let $V_{l,u}$ be the set of visits that user u has made to the location l. Thus, $c_{l.u} = |V_{l,u}|$ denotes the number of visits of user u to location l. The proportion of visits l by user u to the total number of visits l is $p_{l,u} = c_{l,u}/c_l$. According to Shannon entropy [42], the location entropy of l is

$$LE(l) = -\sum_{u \in U_l} p_{l,u} \log_2 p_{l.u} \tag{6}$$

The greater the location entropy, the higher the uncertainty of the location and the level of privacy protection. Location entropy reaches a maximum $\log_2 k$, when all users have the same number of visits, where $k = |U_l|$.

3.5. Weighted Standard Deviation Circle. Due to the unevenness of the spatial point distribution pattern and its sparsity, the parameter of the subspace size generated by clustering or partitioning does not fully reflect the quality of the partitioning result, which in turn affects the availability of data. So we introduce a standard deviation circle to solve this problem. In spatial point mode analysis, standard deviation



Node	U_1,U_2,U_3	U_4,U_5,U_6,U_7,U_8
Size	16	49

(a)

Node	U_5,U_6,U_7	U_8,U_2,U_3,U_4,U_1
Size	6	20

(b)

Figure 2: 3-anonymous partition by Hilbert curve.

circle, weighted standard deviation circle, of coordinate x/y is used to describe the discrete trend of spatial distribution [43].

The radius of the standard deviation circle is similar to the standard deviation in classical statistics, describing the spatial deviation of the observed points.

The standard deviation circle radius calculation method is

$$r = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \overline{x})^2 + (y_i - \overline{y})^2}{n-2}} \tag{7}$$

where $\overline{x}, \overline{y}$ is the center point of the space, x_i, y_i are the coordinates of the point, and n is the number of points in the space. For two areas with equal size, if the total number of points is the same, then the area with a large standard deviation circle radius has a large spatial dispersion [43]. However, when some attributes of the spatial point itself can affect the degree of spatial dispersion, the result of the standard deviation circle will be biased. In order to correct this offset, some attributes of the spatial point can be used as weights to generate a weighted standard deviation circle.

Then the radius is calculated as follows:

$$r_w = \sqrt{\frac{\sum_{i=1}^{n} w_i (x_i - \overline{x})^2 + \sum_{i=1}^{n} w_i (y_i - \overline{y})^2}{\sum_{i=1}^{n} w_i}} \tag{8}$$

where w_i is the attribute weight; other parameters are defined as above (7).

4. Our Proposal

In this chapter, we describe how to design a differential privacy publishing method for a trajectory dataset. First, we introduce our method and then decompose the algorithm into two subalgorithms and finally prove that the whole algorithm satisfies ε-differential privacy.

4.1. Preview. We solve processing the original data by dividing the location set of the trajectory database at each timestamp. For example, the similar points in the original dataset are divided into the same area, and the divided areas are guaranteed to be as accurate as possible. In this way, we can greatly reduce the set of location points within the timestamp, thereby reducing the output domain of the trajectory dataset. Further, similar trajectories are also merged, which greatly increases their counts, reduces the injected noise, and improves data availability.

4.1.1. Differential Privacy Spatial Division Algorithm. This algorithm uses the Hilbert curve to divide the set of location points L_i of each timestamp t_i and divides the original set into multiple subsets, which is regarded as a kind of partition. When using multiple Hilbert curves for dividing, many partitions are produced. The size of the area produced by each partition is different, and the distribution of internal location points is not same. According to the location entropy of each point in an area, the weighted standard deviation circle of all points in the area is calculated, and then the exponential mechanism is used to output the best partition with higher availability. Thus, the location set L_i in the original dataset becomes $\widetilde{L_i}$, as an input to the next algorithm.

4.1.2. Differential Private Data Publishing Algorithm for Trajectory Generated. This algorithm generates generalized spatiotemporal trajectories based on the output $\widetilde{L_i}$ of last step at each timestamp t_i. Then we use the Laplace mechanism to publish the noise count of the generalized spatiotemporal trajectory. To improve efficiency and utility, we only focus on the trajectories that exist in the original dataset and design a method to reduce the noise injected.

As in Figure 3, there are 8 trajectories in the original database and the length is 3. We divide the location set at each timestamp t_i in Algorithm 1 and generalize noisy trajectories by Algorithm 4. Then we release the processed trajectory dataset.

The details of two core algorithms are elaborated separately below.

4.2. Differential Privacy Spatial Division Algorithm. In order to reduce the size of the spatiotemporal location set and

Input: original dataset D, privacy budget ε
node amount k of each area in partition
Output: $\widetilde{L} = \{\widetilde{L_1}, \widetilde{L}_2, \cdots \widetilde{L_{|T|}}\}$ after dividing
(1) Initialize L_i. Assign locations at each time t_i of the trajectory dataset to L_i
(2) For each L_i, computing $|L_i|$ and location entropy for l_i in L_i, determining order of Hilbert curve

$$order = \left\lfloor \log_2^{\sqrt{|L_i|}} \right\rfloor$$

(3) Generate four Hilbert curves, for each curve
(4) computing $aver(r_w) = (1/|G|) \sum_{i=1}^{|G|} r_w$ according to $order$ and k
(5) select r_w with probability P
P is proportional to $e^{(\varepsilon \times u(D,r))/2\Delta u}$
(6) Output the optimal partition of L_i at t_i
(7) Output the set $\widetilde{L} = \{\widetilde{L_1}, \widetilde{L}_2, \cdots \widetilde{L_{|T|}}\}$

Algorithm 1: DPSD(D, ε).

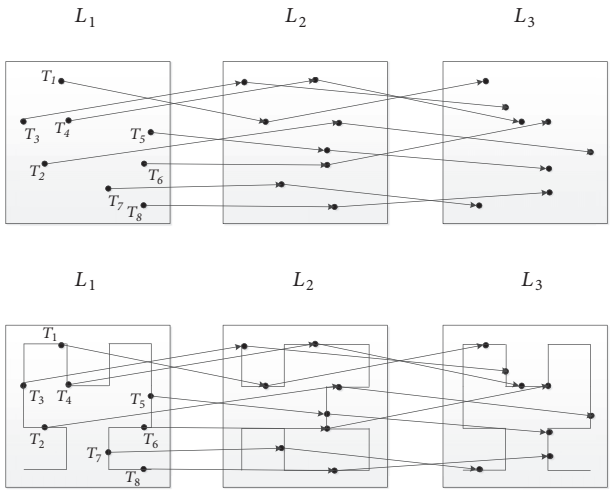


Figure 3: Original trajectory dataset.

ensure the availability of the trajectory data after the differential privacy mechanism, we try to process the location set in the original dataset by dividing the regions and merging them. Hua J [44] pointed out that the k-means method can be used for clustering in the location set, but the traditional k-means method needs to determine the number of clusters k in advance. And the selection of the initial center point of k-means cluster has a great influence on clustering effect and time. For the problem of uneven distribution of spatial points, although density-based clustering method [45] can perform adaptive clustering without determining special the number of clusters, in density-based clustering, the distance between cluster nodes and cluster center is uncontrollable [46], which is not suitable for the application scenario of this paper. So, to solve the above problems, we use the idea of grid partitioning aggregation in [46] and propose a differential privacy spatial division algorithm.

In the original dataset of Figure 3, each trajectory is recorded as T_i; we divide each spatiotemporal point on the trajectory into different location set L_i by time t_i and perform the same dividing operation for each L_i separately. At first, we want to divide the location set as what we introduced in Section 3.3 by average size of cells. However, in the subsequent research, we found the shortcomings of this approach. Consider the division of the following two location distribution.

In Figure 4, a and b represent two kinds of distribution. There is one point in 0-3 cell in a, but there is one point in cells 0 and 1 and two points in cell 3 in b. When dividing, these two distributions will get the same dividing area with same anonymity effect but different service quality because of its uneven density.

For the same reason, in the distribution of c and d, after dividing, the size of the cell area in c is 4, while in d it is 6. But, in fact, the division in d should be better. These two comparative examples in Figure 4 illustrate that Hoa Ngo's dividing method based on the number of grids proposed in [36] is prone to be inaccurate for some distributions, so we recommend using a more accurate method to divide location set, which can indicate the difference in distribution.

We use the standard deviation circle in Section 3.5 to solve this problem. According to formula (7), we can calculate the standard deviation circle radius of the two comparative sample pieces of data in Figure 4. The results are a=1>b=0.935 and c=1.581>d=1.17, which is in line with our observation and actual condition. So it can be used as an evaluation criterion for area dividing. However, we cannot just simply use multiple Hilbert curves for spatial dividing and then choose the best partition with the smallest average standard deviation circle radius, even if this method can produce the most accurate division. Because the optimal partitioning produced by each Hilbert curve is different, this simple way of choosing the minimum output is susceptible to speculative attacks. To solve this problem, we introduce the

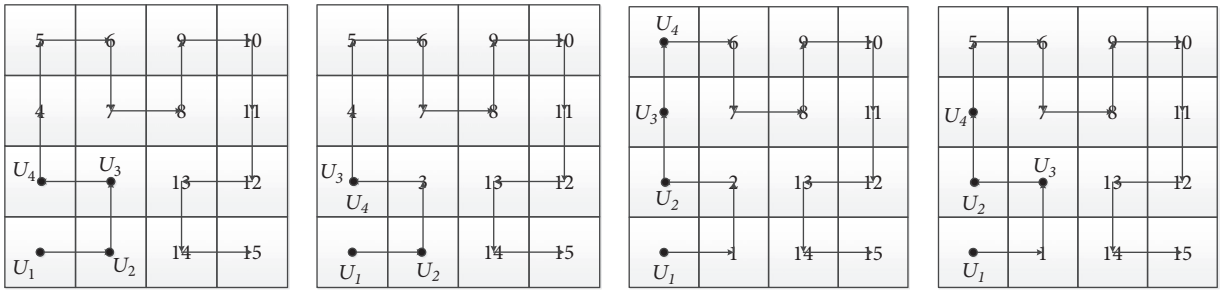


Figure 4: 2-order Hilbert curve partition.

exponential mechanism of differential privacy to ensure a relatively accurate division to be output. The utility function $u(D, r) \longrightarrow R$ is designed as

$$u(D, r) = 1 - \frac{Size(r)}{MaxSize(R)} \quad r \in R \tag{9}$$

where r is the average radius of the standard deviation circle of each area in the partition and R is the collection of all r.

For example, we use three Hilbert curves to divide a location set, and the results are r_1=5, r_2=8, and r_3=10, respectively; then their corresponding utility scores are $u(D, r_1) = 0.5$, $u(D, r_2) = 0.2$, and $u(D, r_3) = 0$. In this way, we get the highest score for the optimal result of dividing the region. Then we can randomly select an output r with the probability $P(D, r)$ proportional to $e^{(\varepsilon \times u(D,r))/2\Delta u}$, where

$$P(D, r) = \frac{e^{(\varepsilon \cdot u(D,r))/2\Delta u}}{\sum_{r \in R} e^{(\varepsilon \cdot u(D,r))/2\Delta u}} \tag{10}$$

ε is privacy budget and Δu is sensitivity of $u(D, r) \longrightarrow R$. Differential privacy requires global sensitivity to be as small as possible to minimize the noise injected. The sensitivity of our utility function is $\Delta u = \max_{r \in R} |u(D, r) - u(D', r)| = 1$, so it can be used. In addition, we noticed that when using data-relative publishing method, the properties of the data itself will also affect the publishing results. Based on this idea, we can replace the standard deviation circle used in the utility function with the weighted standard deviation circle. The attribute of the location entropy in dataset is taken into account, which can be calculated by formula (8) to get weighted standard deviation circle radius. At same time, the utility function should be modified to

$$u(D, r_w) = 1 - \frac{Size(r_w)}{MaxSize(R)} \quad r_w \in R \tag{11}$$

After the process of Algorithm 1, L_i in the original dataset can be divided into different area, as shown in Figure 5, and the pseudocode of the algorithm is shown in Algorithm 1.

The order we use to generate the Hilbert curve is based on the number of location points in the target area. Our aim is that, in the best case, each location can get a Hilbert value, so we need at least n cells, which require the order of Hilbert curve to be $order = \lfloor \log_2^{\sqrt{|L_i|}} \rfloor$.

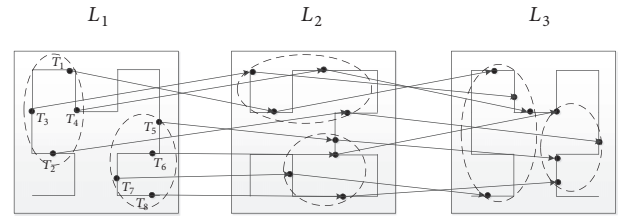


Figure 5: Divided original dataset.

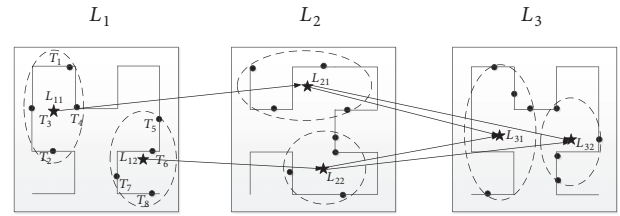


Figure 6: Replaced trajectory dataset.

4.3. Differential Private Data Publishing Algorithm for Trajectory Generated. Through Algorithm 1 in previous section, location set L_i at each timestamp t_i of the original spatiotemporal trajectory dataset is divided into $\widetilde{L_i}$. After this, in order to construct generalized trajectories that are similar to the trajectories in the original dataset, we also need to perform the Algorithm 4. It is to generate a trajectory dataset $\overline{D}$ to be published by divided location sets $\widetilde{L_i}$. It should be noted that, in Algorithm 1, once k is fixed, the number of partitions in $\widetilde{L_i}$ is determined to be $[|L_i|/k]$. If the trajectory's length is l, the number of all possible trajectories is $[|L_i|/k]^l$. Suppose that the number of partitions is 80 in each $\widetilde{L_i}$; the total number of possible trajectories with the length 36, like our setting in experiment, which we have to consider is 80^{36}. It is obviously unacceptable if we add Laplace noise on all of these trajectories. Therefore, we need to process the output of Algorithm 1 to reintegrate the different sets of location partitions into trajectories that are similar to original dataset.

We adopt the following strategy: first, for each $\widetilde{L_i}$, location points that belong to the same area are replaced by the center of this area, and other points in the area are deleted, so that each area corresponds to only one point. Then, according to

Table 2: Generalized trajectories.

Generalized Tra	Original Tra	Count
$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{31}$	T_1, T_3, T_4	3
$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{32}$	T_2	1
$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{31}$	NULL	0
$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{32}$	NULL	0
$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{31}$	NULL	0
$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{32}$	NULL	0
$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{31}$	T_7	1
$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{32}$	T_5, T_6, T_8	3

every trajectory in the original dataset, we find the area it passed and replace this original location point with center of this area. Finally, we get generalized $\widetilde{D}$ trajectory dataset after dividing, as shown in Figure 6.

After the replacement, the count of some generalized trajectories also increases because each divided subset contains multiple location points, as can be seen from Table 2.

Compared to the unprocessed trajectories, increasing the count of trajectories reduces the disturbance of adding noise to the smaller count, which allows us to publish trajectory data more accurately than before. For example, if the original dataset is not processed, its published dataset is $D = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8\}$; the count set of trajectories is $C = \{1, 1, 1, 1, 1, 1, 1, 1\}$. We will have very low availability after adding Laplace noise on such a sequence. The generalized dataset after replacing is $\widetilde{D} = \{\widetilde{T_1}, \widetilde{T_2}, \widetilde{T_3}, \widetilde{T_4}\}$, where each $\widetilde{T_i}$ is the generalized trajectory in Table 2 and $\widetilde{D}$ is the noisy dataset that contains all generalized trajectories.

The count set is $C = \{3, 1, 1, 3\}$. Such a trajectory dataset with differentiated count can withstand the influence of Laplace noise and achieve better usability.

We have noticed that when the number of generalized trajectories is large, the total amount of Laplace noise can also be large. Let us take Table 2 as an example; when we add Laplace noise to $\widetilde{D} = \{\widetilde{T_1}, \widetilde{T_2}, \widetilde{T_3}, \widetilde{T_4}\}$, we need to add independent noise N, $N \sim Lap(\Delta f/\varepsilon)$, where $\Delta f = 1$, because any adjacent trajectory dataset's count values differ by up to 1. For these four trajectories in dataset, the noise amount we need is $4N$. But when we sort and group the count set C, $\{3, 1, 1, 3\} \Longrightarrow \{\{1, 1\}, \{3, 3\}\}$.

For the dataset after group, we just need to add $2N$ Laplace noise to satisfy differential privacy, which means that we just add $1/2N$ noise to each trajectory. While when reducing Laplace noise, new errors arise, which we call mean error, because we need to use the average value to restore the count of each trajectory in each group when publishing data. The difference between average count and the true count results in this error. We still use the example $\widetilde{D}$: the count set of dataset $\widetilde{D}$ is $C = \{3, 1, 1, 3\}$. The count set of neighbor dataset of $\widetilde{D}$ may be $C' = \{3, 1, 1, 4\}$. When we add Laplace noise directly, the noise error is $4N$ and mean error is 0. After grouping this count set, C becomes $C' = \{\{1, 1\}, \{3, 4\}\}$; then the Laplace noise is $2N$ and mean error is $|3 - 3.5| + |4 - 3.5| = 1$. If $4N$ is greater than $2N+1$, then we believe that the error generated by the dividing is acceptable and this release method is preferable. From a global perspective, as the number of group increases, the Laplace error increases gradually, and the mean error decreases. They are mutually constrained. However, from the perspective of data availability, we only need to minimize the total error of data release.

Definition 8 (total error). For count set of the trajectory dataset used in this paper, the total published error is the sum of Laplace error and the mean error.

$$Error\,(C) = Error\,(Lap) + Error\,(Mean) \tag{12}$$

$Error(Lap) = \sum_k N$, where k is group size after dividing and N is noise amount.

$$Error\,(Mean) = \sum_{i=1}^{|G|} \left| c\,(T_i) - aver\,(G) \right|, \tag{13}$$

where $c(T_i)$ is count value of every trajectory in group and $aver(G)$ is the average count in group G.

There are two main factors that affect the performance of this release method: (i) the sorting process may not guarantee differential privacy and (ii) the existence of statistical distance outlier may affect usability. For problem (i), when sorting the trajectory count value, sorted partition is sensitive to adding or deleting a single trajectory. We use the same example to illustrate it: $C = \{3, 1, 1, 3\}$ is the count set of $\widetilde{D}$, which is divided into $C = \{\{1, 1\}, \{3, 3\}\}$. The count set of neighbor dataset of $\widetilde{D}$ may be $C' = \{3, 1, 0, 3\}$ and produce $C' = \{\{1\}, \{3, 3\}\}$. It is apparent that the probability that $C' = \{\{1\}, \{3, 3\}\}$ is equal to $C = \{\{1, 1\}, \{3, 3\}\}$ is 0, which could not satisfy differential privacy. Therefore, in order to satisfy the differential privacy, when sorting and dividing the count set, we add Laplace noise to real count and then sort count. By doing this, Laplace mechanism can still make $c(T) + Lap(1/\varepsilon)$ output the same result as $c(T)' + Lap(1/\varepsilon)$ with a high probability when the count value of some trajectory turns from $c(T)$ to $c(T)'$. The sort set not only can provide differential privacy but also is close to real order to a large extent.

The algorithm is designed as shown in Algorithm 2.

In Algorithm 2, ε is the privacy budget required for sorting, but since line (5) replaces the noisy count with the real count and outputs an approximate order of the true values, there is no impact on the published results, so the algorithm does not actually consume the privacy budget [47].

For problem (ii), the influence of the statistical distance outlier trajectory is important. In this paper, the statistical distance outlier trajectory is defined as follows.

Definition 9 (statistical distance outlier trajectory). Given real number δ, T is a trajectory; if the difference between count value $c(T)$ of T and count value of trajectories T_i is d and if the difference $d = |c(T) - c(T_i)| > \delta$, trajectory T is the statistical distance outlier trajectory.

It can be known from the definition that if the outlier trajectory is divided into the same group with nonoutlier

Input: generalized dataset $\widetilde{D}$, privacy budget ε
Output: approximate sorting set of trajectory count *SortSet*
(1) Initialize *SortSet*=null;
(2) For each trajectory in $\widetilde{D}$
(3) Count real count to get C
(4) For each in C
Adding Laplace noise $Lap(1/\varepsilon)$ to real count and get $\widetilde{C}$
(4) Sort the counts in $\widetilde{C}$
(5) Keep the order of sort in $\widetilde{C}$, replace the noise count with original count
(6) return *SortSet*=$\widetilde{C}$

Algorithm 2: CountSort.

trajectories, the mean error will be very large, which will seriously affect the release result. So, we need to first find the outlier trajectory and try to separate the outlier trajectories into a single group to avoid the influence of other mean errors of group.

Generally, we can take the following methods to divide. First, the output of Algorithm 2 is an approximate correct trajectory count ordered set. Supposing that the generalized count set $C = \{c(T_1), c(T_2), c(T_3), \cdots c(T_n)\}$, we calculate the difference between adjacent elements of this approximate order set $d_i = |c(T_{i+1}) - c(T_i)|$ and get the set of difference $d = \{d_1, d_2, d_3, \cdots d_{n-1}\}$. Then we choose the max $d_{\max} = d_i$; by doing this, we can find the maximum difference trajectory in C, which is the relative outlier trajectory. We divide the count set C into $C_1 = \{c(T_1), c(T_2) \cdots c(T_i)\}$ and $C_2 = \{c(T_{i+1}), c(T_{i+2}) \cdots c(T_n)\}$. If the error after dividing $Error(C_1) + Error(C_2) < Error(C)$, we accept this division and then recursively perform the above process on $C = \{\{C_1\}, \{C_2\}\}$ until all subdivisions do not meet the requirements.

However, the process above does not guarantee privacy. When an attacker has strong background knowledge, it is possible to obtain user privacy by speculating attacks at each division. Therefore, in order to protect privacy from the strongest attacker, we use differential privacy to defend. When we choose $d_{\max}$ from $d = \{d_1, d_2, d_3, \cdots d_{n-1}\}$, exponential mechanism can be used to output $d_{\max}$ with a high probability. We design a simple and effective utility function: $u(D, d_i) = d_i$; the bigger d_i is, the higher its score and probability are, which conform the rule. The sensitivity $\Delta u = 1$ because the max difference is 1 when there is only one different record on neighbor dataset. The probability of output d_i is

$$P(D, d_i) = \frac{e^{(\varepsilon \cdot u(D, d_i))/2\Delta u}}{\sum_{d_i} e^{(\varepsilon \cdot u(D, d_i))/2\Delta u}}, \quad d_i \in d \tag{14}$$

We can get correct group division after above steps.

In Algorithm 3, $Error(C)$ is the initial min error and $Error(C) = N + \sum_{i=1}^{n} |c(T_i) - aver(C)|$, where N is Laplace noise and C is the whole count set. The algorithm continually iterates until the global error is no longer reduced, so the number of iterations and partitions cannot be determined.

Input: *SortSet* output by Algorithm 2, privacy budget ε
Output: divided trajectory count set C
(1) Initialize $Error(C)$
(2) Computing difference set d according to *SortSet*
(3) $\varepsilon = \varepsilon/2$, select d_i with the probability proportional to $e^{(\varepsilon \cdot u(D, d_i))/2\Delta u}$, and group C into $C = \{C_1, C_2\}$
(4) if $Error(C_1) + Error(C_2) < Error(C)$
save C_1, C_2, d=$d - \{d_i\}$,
repeat step (3) and (4) on C
(5) else (stop dividing)
(6) return C

Algorithm 3: TraDivision.

Table 3: Published trajectory dataset.

No.	Tra ID	Generalized Tra
1	$\widetilde{T}_1$	$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{31}$
2	$\widetilde{T}_1$	$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{31}$
3	$\widetilde{T}_1$	$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{31}$
4	$\widetilde{T}_2$	$\widetilde{L}_{11} \longrightarrow \widetilde{L}_{21} \longrightarrow \widetilde{L}_{32}$
5	$\widetilde{T}_3$	$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{31}$
6	$\widetilde{T}_4$	$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{32}$
7	$\widetilde{T}_4$	$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{32}$
8	$\widetilde{T}_4$	$\widetilde{L}_{12} \longrightarrow \widetilde{L}_{22} \longrightarrow \widetilde{L}_{32}$

In this algorithm's line (3), $\varepsilon = \varepsilon/2$ means that, in each iteration, the differential privacy mechanism consumes half of the remaining privacy budget in previous step. Because we do not know the exact running times of the process, we can limit the whole budget privacy less than ε.

Both Algorithms 2 and 3 are just part of the differential privacy data publishing algorithm for trajectory generated. The complete algorithm for trajectory generated is as shown in Algorithm 4.

After processing by Algorithm 4, the format of the published trajectory dataset is as Table 3.

4.4. Privacy Analysis. In this section, we analyze the privacy attribute of the algorithm proposed in this paper. The main algorithm of this paper consists of two parts. The first part

Input: $\widetilde{L} = \{\widetilde{L_1}, \widetilde{L}_2, \cdots \widetilde{L_{|T|}}\}$, privacy budget ε
Output: published dataset $\overline{D}$
(1) Initialize $\varepsilon = \varepsilon_1 + \varepsilon_2$, $\overline{D} = \Phi$
(2) Based on $\widetilde{L}$, use the centers of each area to replace original trajectories, making generalized $\widetilde{D}$
(3) *SortSet*=CountSort($\widetilde{D}, \varepsilon$)
(4) C=TraDivision($Sortset, \varepsilon_1$)
(5) For C_i in C, the noisy count c of trajectories in C_i
$c = aver(C_i) + Lap(1/|C_i|\varepsilon_2)$;
add trajectories in C_i into $\overline{D}$
(6) return published dataset $\overline{D}$

Algorithm 4: 2DPA-TG($\widetilde{L}, \varepsilon$).

is the differential privacy spatial partition algorithm. When designing the algorithm, we use the exponential mechanism to guarantee differential privacy. The privacy budget we assign to this algorithm is ε_d for each of the location sets of each timestamp. Then we execute algorithm sequentially. According to Theorem 6 (sequential composition), we have the following theorem.

Theorem 10. *The differential privacy spatial partition algorithm guarantees $|T| \cdot \varepsilon_d$-differential privacy across the entire trajectory dataset.*

The second part differential privacy data publishing algorithm for trajectory generated consists of two subalgorithms, in which there are two parts consuming privacy budget: (i) Algorithm 3 TraDivision uses the exponential mechanism to select the maximum adjacent count difference and (ii) Algorithm 4's line (4) uses the Laplace mechanism to add noise on trajectory counts. Algorithm 3 consumes half of the remaining privacy budget ε each time, so the total privacy budget is $\sum_{i=1}^{k}(1/2^i) \cdot \varepsilon_1 < \varepsilon_1$, where k is the total number of iterations when the algorithm stops.

Line (4) of Algorithm 4 consumes a privacy budget ε_2 for one-time consumption. Then we have the following theorem.

Theorem 11. *Differential privacy data publishing algorithm for trajectory generated satisfies ε_p-differential privacy, $\varepsilon_p = \varepsilon_1 + \varepsilon_2$.*

So, as the sequential property of differential privacy, we have the theorem as follows.

Theorem 12. *The protection mechanism we propose in this paper satisfies ε-differential privacy, and $\varepsilon = |T| \cdot \varepsilon_d + \varepsilon_p$.*

5. Evaluation

5.1. Dataset and Environment. The experimental dataset of this paper is published by Microsoft Research Asia, which contains the trajectories of 10,357 taxis in Beijing in a week. Each record in the dataset is like (taxi_id, time, longitude, latitude) and the GPS sample frequency of each taxi ranges from 1s to 10min. In order to fit the algorithm design and facilitate the experiment, we extracted the data from 6:00 to 12:00 in one day, and the sampling frequency of each data is 10min, which means that every spatiotemporal point on the trajectory has 10min interval. After processing the raw dataset, we obtained original experimental data containing 7,369 taxi trajectories.

The experimental hardware environment is Windows 7, Intel Core i5 6500 CPU, 3.2 GHz, and 8G memory, and the experiment is programmed by Java.

5.2. Utility Metric. The algorithm proposed in this paper causes the loss of usability between the published dataset and the original dataset in two aspects. The first is the loss caused by division and replacement of the location points at each timestamp and the second is the error caused by Laplace noise and exponential perturbation. Therefore, in this section, we evaluate availability of the published trajectory dataset by Hausdorff distance [48] and spatiotemporal range query [49].

Hausdorff distance is a way to measure the degree of similarity between two point sets. The spatiotemporal trajectory can also be seen as a form of point set, so the Hausdorff distance measurement can be used.

The Hausdorff distance between T_i and T_j is calculated by the following formula:

$$HD\left(T_i, T_j\right) = \max\left(hd\left(T_i, T_j\right), hd\left(T_j, T_i\right)\right)$$

$$hd\left(T_i, T_j\right) = \max\left(\left(\min\left\|p_i^m, p_j^n\right\|, \ p_j^n \in T_j\right), \ p_i^m \in T_i\right) \tag{15}$$

$hd(T_j, T_i)$ is the same as above. Obviously, the smaller the Hausdorff distance, the higher the usability.

We use the DPR algorithm in [44] as a comparison experiment. DPR is similar to the algorithm framework of our proposal. The DPR algorithm uses k-means clustering for each timestamp and finally publishes the dataset with Laplace noise and generated fake trajectories. It should be noted that the DPR algorithm needs to specify k as the number of cluster clusters. In our paper, the required k is the minimum number of location points in each partition area.

We calculate the Hausdorff distance between the original trajectory dataset and the published trajectory dataset. In Figure 7, at first, as ε increases, the Hausdorff distance tends

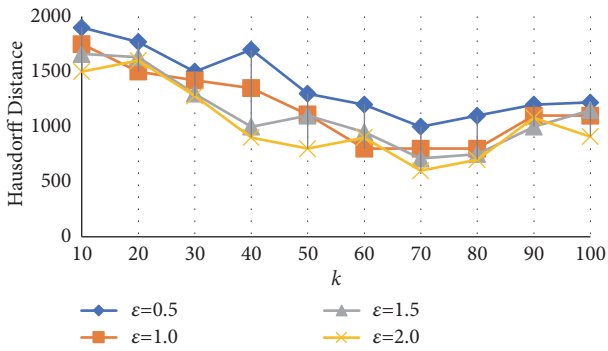


Figure 7: Hausdorff distance varies with k.

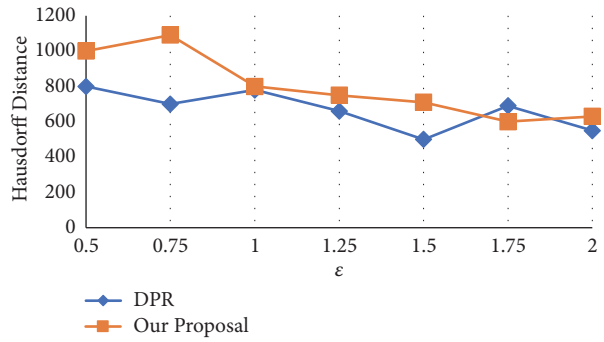


Figure 8: DPR versus our proposal in Hausdorff distance.

to decrease. The reason is that when ε increases, Laplace noise and exponential noise will decrease, which make the difference between published dataset and the original dataset smaller. With the increase of k, Hausdorff distance decreases first and then increases. This is because k has an influence on the location set L_i in Algorithm 1 and number of generalized trajectories in Algorithm 4. After the value of k is too large, the partition area is increased, and the distance between center of L_i and other points in L_i is increased, thereby affecting the availability of the entire dataset. When k=70, the Hausdorff distance gets its minimum, so we fix k=70 to perform the subsequent experiments.

We experimentally compared the trajectory dataset released by the DPR algorithm with the trajectory dataset published by our algorithm. It can be seen from Figure 8 that the Hausdorff distance of the DPR algorithm is a little smaller than our algorithm in most cases. This is because the DPR algorithm uses k-means based clustering algorithm when dealing with location point sets at each timestamp. Such algorithms tend to choose nearest neighbors as the same category, with higher accuracy in Euclidean distance based comparison method. But considering that the time complexity of k-means algorithm is $O(knmt)$ and the DPR is $O(n^2)$, while our proposal is $O(n \log n)$, it is perfectly acceptable to sacrifice a small amount of Hausdorff distance in exchange for time complexity.

Spatiotemporal range query is a method for measuring trajectory data quality proposed in [49]. In the experiment, we, respectively, use the algorithm proposed by this paper and DPR algorithm to publish dataset $\overline{D}$ and then perform two kinds of spatiotemporal range query on D and $\overline{D}$. We calculate the distortion of the query results of the two algorithms as follows:

$$loss\left(D, \overline{D}\right) = \frac{\left|Q\left(D\right) - Q\left(\overline{D}\right)\right|}{\max\left(Q\left(D\right), Q\left(\overline{D}\right)\right)} \tag{16}$$

For better comparison with the existing work, we choose two types of spatiotemporal range queries, namely, *PSI* query and *DAI* query.

$PSI(Possibly\_Sometimes\_Inside(T, R, t_s, t_e))$ query is what might happen in a certain period of time, which means count trajectory T that might appear in the area R during time $[t_s, t_e]$.

$DAI(Definitely\_Always\_Inside(T, R, t_s, t_e))$ query is what must happen in a certain period of time, which means count trajectory T that definitely appears in the area R during time $[t_s, t_e]$.

We generate two sets of queries Q_1 and Q_2:

$$Q_1 = selectcount(*)\ from\ D\ where\ PSI(T, R, t_s, t_e)$$
$$Q_2 = selectcount(*)\ from\ D\ where\ PSI(T, R, t_s, t_e)$$

According to the parameter settings in [23], we set R to a circular area of 1000 m and 500 m and the time interval $[t_s, t_e]$ is two hours. We generate 1000 queries and the experimental results are shown in Figure 9.

These experiments verified the degree of distortion of *PSI* and *DAI* query when the radius of the area is 500 m or 1000 m. As can be seen from Figure 9, when the privacy budget ε increases, the distortion of both algorithms will gradually decrease, which is in line with our intuitive speculation and actual situation, because, with the increase of privacy budget ε, the degree of privacy is reduced and the injection noise is also reduced. Then data will be closer to the real data; thus the usability is improved. Take Figure 9(a) as an example, when we perform this kind of query $select\ count(*)\ from\ D\ where\ PSI(T, R, t_s, t_e)$ on $\overline{D}$ and D, with the increasing of privacy budget ε, the query distortion of our algorithm is gradually reduced and is always smaller than the distortion of the DPR algorithm.

Through the results in Figure 9, we can see that the data publishing algorithm proposed in this paper is stronger than the DPR algorithm in spatiotemporal range query, because, in order to ensure sufficient privacy, the DPR algorithm uses a fake trajectory-based method to add fake trajectories into published dataset. Even though the DPR's k-means clustering stage is better than our algorithm, there are not only the Laplace noise but also the disturbance error caused by fake trajectory, while our algorithm has no such drawbacks. In Figure 10, we compare the time efficiency between DPR and our method. We implement the experiment on parameter k, which is the location number in each partition. As k increases, the time consumed in process decreases simultaneously, because, with k growing, the partition in each timestamp will decrease, which make generalized trajectories less. So, the time based on sort and group these trajectories will be less than before. The result shows that our method is better

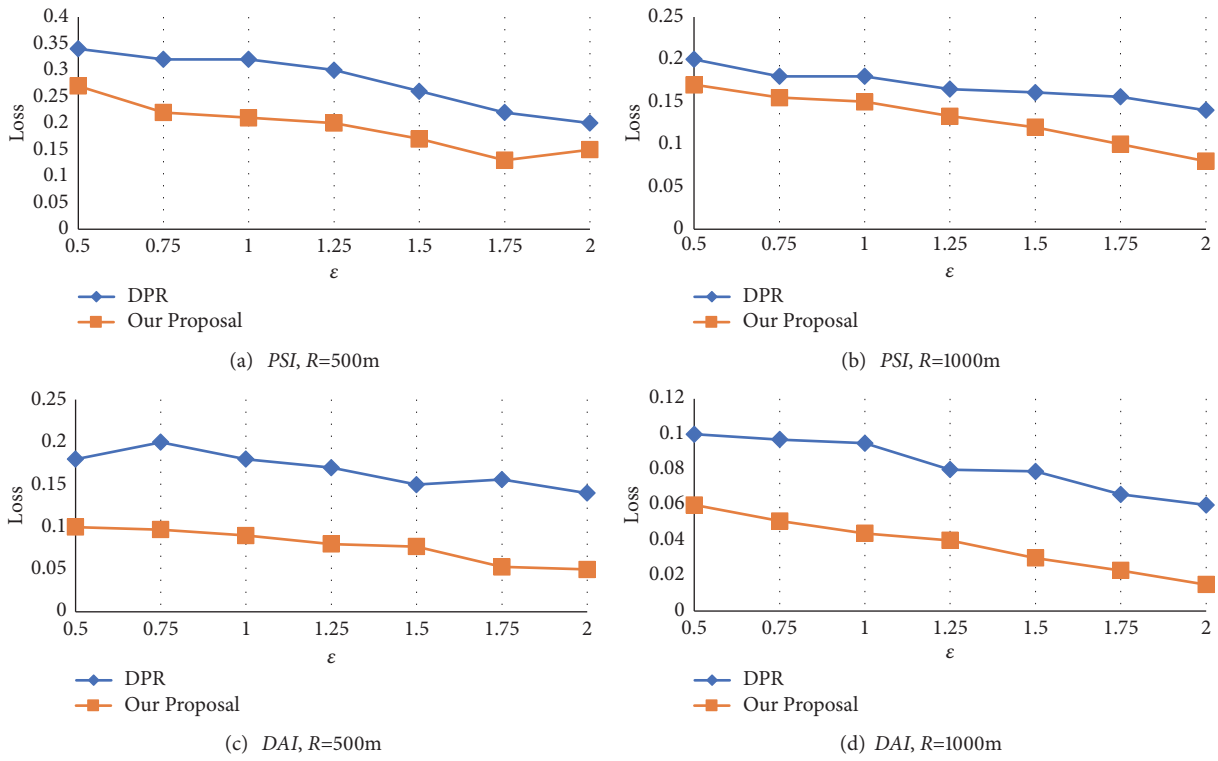


Figure 9: Query results of *PSI* and *DAI* with R=500m and R=1000m.

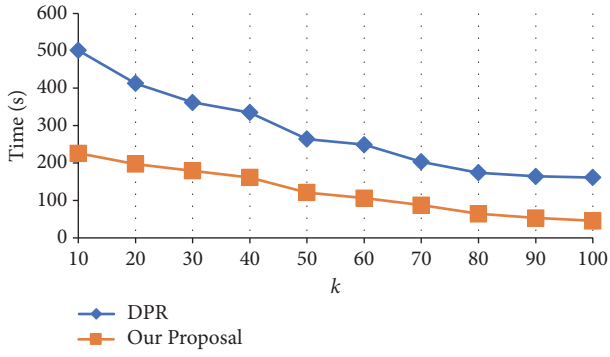


Figure 10: DPR versus our proposal in time efficiency.

than DPR method because our time complexity is $O(n \log n)$ as we mentioned before. Therefore, from comprehensive effect considerations, performance of our method on these experiments proves that our proposal is more practical and better than the DPR algorithm.

6. Conclusion

In this paper, we propose a spatial partition based trajectory dataset publishing algorithm. This algorithm satisfies differential privacy with high utility and can run in less time than existing method. The algorithm uses exponential mechanism to output more accurate location point partitions and trajectory count group to ensure privacy and then release data after adding Laplace noise into trajectory count. According to our knowledge, this is the first paper to use Hilbert curve to divide amount of trajectories in a noninteractive way. At last, we perform spatiotemporal range queries on real trajectory dataset, and the results are better than existing algorithm. Besides, the published data can achieve smaller Hausdorff distance within the tolerance. The two experiments show that our method can be used in practice effectively. Our method deserves continuous study in the future: before running the algorithm, we preprocess the raw dataset to get the original dataset with same time interval. In fact, the collected GPS information will not be completely regular, and the sampling interval may change randomly or even have an interruption. So how to publish such irregular dirty datasets is a hard problem we need to study in our future work.

Data Availability

The trajectory data used to support the findings of this study can be downloaded from https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/ and the detailed instructions can be found in https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/User_guide_T-drive.pdf.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This paper is a fundamental research and is supported by Funds for the Central Universities (Grant no. HEUCF180603) and Harbin Application Technology Research and Development Project (Grant no. 2016RAQXJ063 and Grant no. 2016RAXXJ013).

References

[1] H. Zheng and M. Xiaofeng, "Research on trajectory privacy protection technology," *Journal of computer science*, vol. 34, no. 10, pp. 1820–1830, 2011.

[2] Y. Huo, C. Yong, and Y. Lu, "Re-ADP: real-time data aggregation with adaptive w-event differential privacy for fog computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 6285719, 13 pages, 2018.

[3] K. Zhang, Q. Han, Z. Cai, and G. Yin, "Rippas: A ring-based privacy-preserving aggregation scheme in wireless sensor networks," *Sensors*, vol. 17, no. 2, pp. 1–19, 2017.

[4] Q. Han, S. Liang, and H. Zhang, "Mobile cloud sensing, big data, and 5G networks make an intelligent and smart world," *IEEE Network*, vol. 29, no. 2, pp. 40–45, 2015.

[5] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science & Engineering*, vol. 99, 2018.

[6] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Communications Magazine*, vol. 5, no. 8, pp. 33–43, 2018.

[7] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart IoT systems: a consideration from privacy perspective," *IEEE Communications Magazine*, vol. 10, no. 2, pp. 12–20, 2018.

[8] Z. Cai, Z. He, and X. Guan, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable & Secure Computing*, p. 99, 2016.

[9] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, p. 99, 2017.

[10] X. Zheng, Z. Cai, G. Luo, L. Tian, and X. Bai, "Privacy-preserved community discovery in online social networks," *Future Generation Computer Systems*, 2018.

[11] Z. He, Z. Cai, and X. Wang, "Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks," in *Proceedings of the 35th IEEE International Conference on Distributed Computing Systems (ICDCS '15)*, pp. 205–214, July 2015.

[12] Q. Han, D. Lu, K. Zhang, X. Du, and M. Guizani, "Lclean: a plausible approach to individual trajectory data sanitization," *IEEE Access*, vol. 6, pp. 30110–30116, 2018.

[13] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM '17)*, pp. 1–9, Atlanta, GA, USA, May 2017.

[14] L. Sweeney, "*k*-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[15] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "*L*-diversity: privacy beyond *k*-anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, 2007.

[16] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: privacy beyond k-anonymity and *l*-diversity," in *Proceedings of the 23rd International Conference on Data Engineering*, pp. 106–115, IEEE, Istanbul, Turkey, April 2007.

[17] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith, "Composition attacks and auxiliary information in data privacy," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 265–273, August 2008.

[18] R. C. Wong, A. W. Fu, K. Wang, P. S. Yu, and J. Pei, "Can the utility of anonymized data be used for privacy breaches?" *ACM Transactions on Knowledge Discovery from Data*, vol. 5, no. 3, pp. 1–24, 2011.

[19] C. Dwork, "Differential privacy," in *Proceedings of the International Colloquium on Automata, Languages, and Programming*, vol. 4052, pp. 1–12, Springer, Berlin, Heidelberg, 2006.

[20] X. Ping, Z. Tianqing, and W. Xiaofeng, "Differential privacy protection and its application," *Journal of Computer Science*, vol. 37, no. 1, pp. 101–122, 2014.

[21] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS '14)*, pp. 251–262, November 2014.

[22] O. Abul, F. Bonchi, and M. Nanni, "Never walk alone: Trajectory anonymity via clustering," ISTI-CNR, Tech. Rep. ISTI, 2007.

[23] O. Abul, F. Bonchi, and M. Nanni, "Never walk alone: Uncertainty for anonymity in moving objects databases," in *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE '08)*, pp. 376–385, April 2008.

[24] Z. Jing and Z. Yuan, "Qinghua LTrajectory privacy preserving method based on trajectory frequency suppression," *Journal of Computer Science*, vol. 37, no. 10, pp. 2096–2106, 2014.

[25] M. Terrovitis and N. Mamoulis, "Privacy preservation in the publication of trajectories," in *Proceedings of the 9th International Conference on Mobile Data Management (MDM '08)*, pp. 65–72, April 2008.

[26] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of the 2nd International Conference on Pervasive Services (ICPS '05)*, pp. 88–97, IEEE Press, July 2005.

[27] N. Mohammed, B. C. M. Fung, and M. Debbabi, "Walking in the crowd: Anonymizing trajectory data for pattern analysis," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*, pp. 1441–1444, Hong Kong, China, November 2009.

[28] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: Optimal strategy against localization attacks," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '12)*, pp. 617–626, October 2012.

[29] G. Theodorakopoulos, R. Shokri, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Prolonging the hide-and-seek game: optimal trajectory privacy for location-based services," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society (WPES '14)*, pp. 73–82, 2014.

[30] R. Chen, B. C. M. Fung, and C. B. Desai, "Differentially private trajectory data publication," *Computer Science*, 2011.

[31] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the 3rd Theory of Cryptography Conference (TCC '06)*, vol. 3876, pp. 363–2385, New York, NY, USA, March 2006.

[32] F. Mcsherry and K. Talwar, "Mechanism design via differential privacy," in *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pp. 94–103, IEEE Computer Society, 2007.

[33] X. Liu and G. F. Schrack, *An Algorithm for Encoding and Decoding The 3-D Hilbert Order*, IEEE Press, 1997.

[34] L. U. Feng, "A GIS spatial indexing approach based on Hilbert ordering code," *Journal of Computer Aided Design & Computer Graphics*, 2001.

[35] J. Liu, Y. Pan, M. Li et al., "Applications of deep learning to MRI images: A survey," *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 1–18, 2018.

[36] G. S. Yadav and A. Ojha, "A scalable data hiding scheme using hilbert space curve and chaos," in *Proceedings of the IEEE Trustcom/BigDataSE/ISPA*, pp. 905–909, Helsinki, Finland, August 2015.

[37] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1719–1733, 2007.

[38] H. Ngo and J. Kim, "Location Privacy via Differential Private Perturbation of Cloaking Area," in *Proceedings of the 28th IEEE Computer Security Foundations Symposium (CSF '15)*, pp. 63–74, July 2015.

[39] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, 2003.

[40] B. Zhou, J. Li, X. Wang et al., "Online Internet traffic monitoring system using spark streaming," *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 47–56, 2018.

[41] C. G. Peng, H. F. Ding, Y. J. Zhu, Y. L. Tian, and Z. F. Fu, "Information entropy models and privacy metrics methods for privacy protection," *Journal of Software*, vol. 27, no. 8, pp. 1891–1903, 2016.

[42] C. E. Shannon and W. Weaver, "The mathematical theory of communication," *Physics Today*, vol. 3, no. 9, pp. 31-32, 1950.

[43] Z. Zhijie, P. Wenxiang, and Z. Yibiao, "Description and application of discrete trend in spatial point pattern analysis," *Chinese Health Statistics*, vol. 25, no. 5, pp. 470–473, 2008.

[44] J. Hua, Y. Gao, and S. Zhong, "Differentially private publication of general time-serial trajectory data," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks (IEEE INFOCOM '15)*, pp. 549–557, May 2015.

[45] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 169–194, 1998.

[46] N. Adrienko and G. Andrienko, "Spatial generalization and aggregation of massive movement data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 2, pp. 205–219, 2011.

[47] Q. Han, B. Shao, L. Li, Z. Ma, H. Zhang, and X. Du, "Publishing histograms with outliers under data differential privacy," *Security and Communication Networks*, vol. 9, no. 14, pp. 2313–2322, 2016.

[48] C. Yanjun, *Research on Clustering Algorithm for Mass Trajectory Data*, Beijing Jiaotong University, 2015.

[49] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain, "Managing uncertainty in moving objects databases," *ACM Transactions on Database Systems*, vol. 29, no. 3, pp. 463–507, 2004.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 1265352, 11 pages
https://doi.org/10.1155/2018/1265352

Research Article

Privacy Preservation for Friend-Recommendation Applications

Weicheng Wang,[1] **Shengling Wang**,[1] **and Jianhui Huang**[2]

[1]*College of Information Technology and Science, Beijing Normal University, Beijing, China*
[2]*Institute of Computing Technology, The Chinese Academy of Sciences, Beijing, China*

Correspondence should be addressed to Shengling Wang; wangshengling@bnu.edu.cn

Received 1 August 2018; Accepted 9 October 2018; Published 21 October 2018

Guest Editor: Yan Huo

Friend-recommendation applications as one kind of typical social applications can satisfy the social contact needs of different users and become tools for developing a social relationship. However, the privacy leakage has turned into an insurmountable obstacle to the market success of such applications. Existing privacy protection approaches for social applications either introduce untrusted third parties or sacrifice information accuracy. As for friend-recommendation applications particularly, the multihop trust chain and anonymous message methods still have a defect that the hacker can act as a user to acquire information. In this paper, we put forward the privacy protection mechanism based on zero knowledge without any privacy leakage to the application server. In detail, the server knows nothing about the user's information, but can still provide users with accurate information on friend recommendation. We also analyze the potential attack methods and propose the corresponding solution. Our simulation results verify the effectivity and efficiency of our scheme.

1. Introduction

With the growing popularity of smart-devices, such as mobile phones, laptops, and tablets, the social applications gradually play an essential role in people's daily life [1, 2]. Facebook has 1.65 billion users with 1 billion active users simultaneously, and Twitter has 600 million users. In China, Ten-cent QQ has 829 million active users, WeChat has over 700 million active users, etc. [3]. The social applications mostly satisfy the social contact needs of different users and become tools for developing a social relationship. They can provide services for people without the limitation of distance. People can find friends who have the same interests and hobbies in the world and communicate whenever and wherever they want.

As one kind of typical social applications, friend-recommendation application has a primary purpose that it helps users to find friends based on the information provided by the users so that the recommended friends can satisfy the requirements of the users. According to the user's interests and location, the application can recommend suitable people to the users. However, once a user sends his query to the server, it is possible that this information is sold to the third party. Furthermore, even if the encryption algorithm is perfect, the malicious people can still act as a user to acquire information from the server. Once a user finds its privacy is leaked by the social applications, he will stop using such applications. When a large number of users stop using them, it is easy to cause the collective panic and the trust crisis.

Existing privacy protection schemes for social applications either introduce untrusted third parties [4–6] or sacrifice information accuracy [7–11]. In the schemes of introducing a third party, the third party acts as the agent of the users to submit the query requests to the server and returns the query results from the server to the users. In this case, the privacy information scattered in various service providers is concentrated in a few third parties. Hence, the risk and scale of privacy leakage are increased sharply. Once the third party is hacked, the crisis of online incredibility will endanger the whole social network heavily. In addition, considering the schemes of sacrificing information accuracy to realize the privacy protection, the server cannot provide perfect services based on the user's blurred private information. Besides, with the help of big data analysis, it is possible that the obscured information can be inferred or restored. Some of the methods used for friend-recommendation applications particularly propose the multihop trust chain [12] or utilize anonymous

message [13, 14]. However, the malicious hackers can still act as a user to acquire information.

Motivated by these, our paper aims to realize zero-knowledge privacy protection. Here, zero knowledge implies that the server knows nothing. Therefore, our aim is challenging, because on the one hand the server needs information from the users to provide services, but on the other hand such information needs to be kept secret from the server.

To realize the seemly conflicted aims, a privacy preservation framework is proposed for friend-recommendation applications, including the communication processes between several entities and data structures. Furthermore, we propose an encryption algorithm based on homomorphic encryption and grouping attributes, so that the users' information can be protected well. We propose a method of district limitation for matching operation which can decrease the possibility of information leakage. We give two possible decryption situations and illustrate the different operations in the two cases. Finally, the potential attack methods are analyzed and the corresponding solutions are proposed. Our simulation results verify the effectivity and efficiency of the proposed scheme.

The remainder of the paper is organized as follows. Section 2 is for the related work. A summary of the framework is presented in Section 3. Section 4 introduces the detailed processes of the algorithms including encryption, decryption, and matching. Some specific defense designs are introduced in Section 5. Section 6 reports the results of our numerical simulations. Our conclusions are summarized in Section 7, followed by the acknowledgment.

2. Related Work

In the social network system, privacy is a rather important issue [15–20]. There are several methods achieving privacy preservation aim, which are based on k-anonymity model. The basic idea of k-anonymity is to remove some features so that each item is not distinguishable among other k items. For data protection, it protects data at the cost of the original data quality [9]. For location-based services (typical social applications), it realizes the privacy protection through blurring user's locations [7, 8, 21].

Based on the k-anonymity, a clustering perturbation algorithm for privacy protection in social networks was proposed [9]. It considers preserving privacy of vertex properties and community structures simultaneously. The algorithm introduces a strategy of exchanging attributes between vertices with the same degree randomly to induce attackers to search for false targets and maintain the whole structure of the network. In 2014, Rongxing Lu et al. [22] proposed a privacy-preserving framework for the local-area mobile social application (PLAM). It employs a privacy-preserving request aggregation protocol with k-anonymity and l-diversity properties, without involving a trusted anonymizer server to keep user's preference private, and integrates unlinkable pseudo-ID technique to achieve user's identity privacy and location privacy.

However, k-anonymity model presents a problem of accuracy loss and attackers can still perform trajectory attacks. It is not suitable in situations where the requirement of information accuracy is extremely high. In addition, k-anonymity does not protect users privacy when they are in a densely populated area.

Besides, a trust-based friend-recommendation scheme used for privacy-preserving was proposed by Linke Guo et al. [12], which applies users' attributes to find suitable friends and establishes social relationships with strangers via a multihop trust chain. Similarly, Squicciarini et al. [23] used game theory to model the privacy management for content sharing. This work can provide automatic access policy generation for users' profile information. In 2015, K. Samanthula et al. [13] utilized the concept of protecting the source privacy through anonymous message routing to recommend friends accurately and efficiently.

In 2018, a data sanitization strategy [24] was proposed which keeps the benefits brought by social network data, while sensitive latent information can still be protected. Even considering powerful adversaries with optimal inference attacks, the proposed strategy can still preserve both data benefits and social structure while guaranteeing optimal latent-data privacy [3]. To resist attacks, SmartMask, a context-based system-level privacy protection solution, was proposed, which was designed to learn users' privacy preferences under different contexts and provide a transparent privacy control for MSN users [25].

Furthermore, because of the importance of photos in friend-recommendation applications, several methods used for mainly protecting the images were proposed. A tool called iPrivacy (image privacy) was developed for releasing the burden from users on setting the privacy preferences when they share their images for special moments. It realizes photo protection by blurring the privacy-sensitive objects automatically [26]. Besides, a concept of changing the granularity of access control from the level of the photo to that of a user's personally identifiable information (PII) was proposed. In this work, it considers the face as the PII. When another user attempts to access a photo, the system determines which faces the user does not have the permission to view and presents the photo with the restricted faces blurred out. However, this mechanism can only be used for photo protection. The scenarios they can be applied on are limited [27].

Moreover, some existing ways of protecting privacy often introduce a third party [4]. All of the keys are stored in the third part. The information should be encrypted before being sent to the server and be decrypted while the server sends it back. However, in the solution of introducing a third party, the privacy information scattered in various service providers will be concentrated. We cannot guarantee the third party is able to protect those keys and the information.

3. Framework

The whole purpose of the proposed framework is to provide a user with the accurate social information according to its query without leaking any query information to the friend-recommendation application system. Query information is

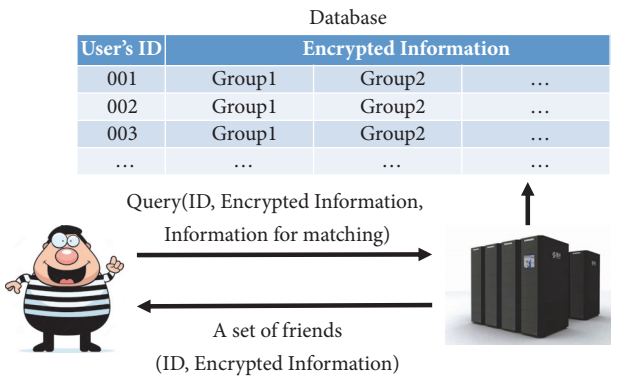


Figure 1: The proposed privacy preservation framework for friend-recommendation applications.

User's ID		
Attributes		Personal Information
Group1	Age	Name
	location	
	…	Personal statement
Group2	Favorite movies	
	Favorite sports	Individual Signature
	…	
…	...	…

Figure 2: The information structure.

often sensitive because it is related to user's privacy. It always contains a user's personal information such as habits, hobbies, and preferences. Hence, our framework aims to realize zero-knowledge privacy protection. Here, *zero knowledge* implies the server knows nothing. Therefore, our aim is challenging, because on the one hand, the server needs information from the user to provide service, but on the other hand, such information needs to be kept secret to the server.

To realize the seemly ambivalent aims, we propose a privacy preservation framework for friend-recommendation application systems. The preservation is described in the following steps. The framework is shown in Figure 1.

(1) The friend-recommendation application server broadcasts three pieces of information to all users: (a) the attributes users can submit; (b) the attribute groups; (c) the part of keys used to decrypt users' information, each of which corresponds to one attribute group's value and hence different attribute groups' values have different keys. The reason why part of keys is related to users' attribute values is that this will make users who have the same attribute groups' values be able to decrypt the corresponding attribute values of the recommended friends.

(2) As required by the server, the user's information like name, age, hobbies, etc. are divided into several groups. For each group, the information is encrypted by homomorphic encryption algorithm. Then user's ID attached to all of the encrypted information will be sent to the server. The key used for encryption consists of two parts. The first part corresponds to the user's attribute values, which is obtained from the broadcast message sent from the server and the other is selected by the user freely. Hence, the key of each attribute group's value is different. Besides, our framework provides two matching patterns: Matching with Strict and Matching with Loose. For the Strict pattern, it requires that all the attribute groups of the friends recommended by the server should match those appointed by the user. As for the Loose pattern, it only requires that part of the attribute groups of the fiends recommended by the server matches those appointed by the user. The user needs to appoint the matching type and specific attribute groups. In the query information, we use 0 to represent the Strict pattern and 1 to represent the Loose pattern. The detailed content will be introduced in the Matching Algorithm section.

(3) Once receiving the information from a user, the server saves the information and finds friends who have the same attributes with the user using the matching algorithm based on homomorphic encryption algorithm in its database. For example, if a user sends his information: age twenties, love watching movies, the algorithm will find people who have nearly the same age and same preference in the database. It is worth noting that because all of the information in the server is encrypted, the user's privacy will be protected.

(4) The server returns the recommended friends' ID and information encrypted to the user. The user can select suitable people and get a connection with them using their ID. Moreover, the user can decrypt part of friends' information.(Through decryption, the user can find out which attributes are the same as those of the recommended friends.) This can be achieved because when a group of attributes of two users is the same, the corresponding keys of two users will be the same. These are related to our matching pattern, especially Loose pattern. How to deal with it will be introduced in the next section in detail.

At Step (1), before a user sends the query information to the server, its information needs to be encrypted first. Actually, the information of the user is divided into two types in our framework. The first one is attributes including hobbies, age, etc. The second type, such as name and personal statement, is merely the personal information. Because of the different characteristics of the information, we program them in different ways. It will be discussed in the Simulation section. The information structure of a user is depicted in Figure 2.

During the query process, there exists one potential privacy leakage issue. Specifically, If a hacker acting as a user asks for recommended friends, he will receive several users' information and know that their information is similar to that of him. In this way, he can acquire users' private information.

To solve this problem, we use users' locations as a special attribute [16] and divide other attributes into groups to reduce the probability of privacy attack from the malicious hackers. With location attributes, the server only recommends friends who are near to users. Grouping attributes make the user be able to only decrypt part of attributes of recommended friends. These methods enable users' privacy to be protected, not only when people want to look for friends, but also when they are recommended friends.

In detail, we separate the area into small equally sized grids whose vertex's longitude and latitude are both integers and transform user's coordinate into the nearest grid vertex. (The reason that the longitude and latitude of a grid's vertex are required to be integers is that it is easy for homomorphic encryption to deal with integers.) Now we introduce the following definition of location.

Definition 1 (a user's coordination). a user's coordination is that of a grid's vertex which is the nearest to the user's current location.

Definition 2 (district). The focal user's coordination is the center of the district which is a square whose length of the side is given. The length of the side of the district is an integral multiple of that of the grid.

In order to reduce the probability of information leakage, the proposed framework only recommends people whose district overlaps that of the user. This requires that the user who sends query should attach his location to the attributes.

For the sake of increasing the security of privacy further, we separate the attributes into several groups. Therefore, when recommending friends to the user, the attributes of two users do not need to be exactly the same, but only part of them. This approach makes sure that when users are recommended, only part of the information of the user will be known. This reduces the probability of leakage of user's privacy to a certain extent.

So far, the user can find friends who are suitable for him. In the whole process, the server operates on the cipher-text. This ensures that the user's private information will not be leaked. Even if a hacker invades the server, there is no chance that he can get any useful information because all the information is encrypted. Besides, even if the server knows the part of the keys corresponding to each attribute group's value combination, it is not easy for the server to decrypt the information of the user. This is because there are too many combinations of the attribute group's value, so it is hard to try all the possible keys to decrypt the information of users. We will further discuss how it is difficult for the server to attack the privacy of users in the Simulation section.

4. Algorithm

In order to realize zero-knowledge privacy protection, our framework adopts algorithms based on homomorphic encryption [28–30]. Homomorphic Encryption allows complex mathematical operations on cipher-texts, generating an encrypted result which matches the result which is operated on the plaint-text and then decrypted [29]. The purpose of homomorphic encryption is to realize a more secure method for data processing. In the following, we detail our algorithms based on homomorphic encryption.

Input:
M: all the attributes groups($M = M_1, M_2....M_k$);
M_i: the i attributes group;
I: personal information ($I = I_1, I_2....I_k$);
I_i: i personal information version;
P_i: the key of i attributes group;
Q_i: the key of i attributes group;
R_i: the key of i attributes group;
K: the number of attributes groups;
Output:
$C([M])$: all the encrypted attributes ($M = M_1, M_2....M_k$);
$C([I])$: all the encrypted personal information ($I = I_1, I_2....I_k$);
(1) **for** $i = 1; i \leq K; i++$ **do**
(2) $N_i \longleftarrow P_i \times Q_i$;
(3) **if** $R_i \neq 0$ **then**
(4) $C([M_i]) \longleftarrow (M_i + P_i \times R_i) \bmod N_i$;
(5) $C([I_i]) \longleftarrow (I_i + P_i \times R_i) \bmod N_i$;
(6) **end**
(7) **end**
(8) return $C([M]), C([I])$;

Algorithm 1: Encryption.

4.1. Encryption. To preserve the privacy of users, the query needs to be encrypted by homomorphic encryption algorithm, shown in Algorithm 1. In order to guarantee the safety of the user's information, as mentioned in the above section, the key used to encrypt information consists of two parts: one corresponds to the attribute groups' value, and the other is selected by the user randomly. Hence, the key of each attribute group is different. Let M_i represent the user's attribute group i, I be the user's personal information, I_i be the i personal information, and K represent the number of attribute groups. The corresponding encryption process is depicted as follows:

For every attribute group i, according to the broadcast information, the client will select two safe large prime numbers P_i and Q_i according to the attribute values of the user. If the users' attribute values are the same, the generated P_i and Q_i are the same. This will make sure that if two users are matched, they can encrypt at least part of the information of each other. The client generate a product $N_i = P_i * Q_i$.

Using the encryption algorithm, the value of each attribute group i is encrypted to the cipher-text $C([M_i])$. $C([M_i]) = (M_i + P_i \times R_i) \bmod N_i$, where R_i is selected by the user randomly to guarantee the information security.

The user's personal information I is encrypted by the key of each attribute groups i apart into several

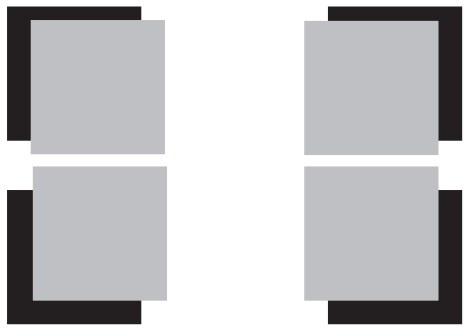


Figure 3: The location matching.

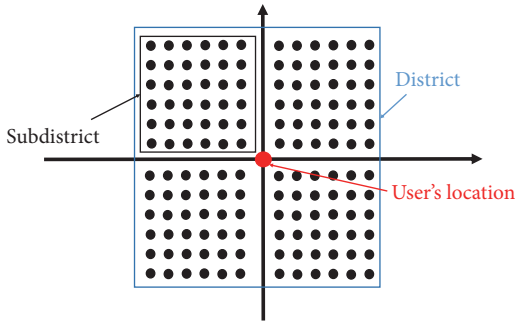


Figure 4: The location structure.

```
k is the number of queue which saves the vertexes.
m^2 is the number of vertexes in the subdistrict.
Input:
La: user_a's information;
Lb: user_b's information;
Output: true or false
(1) result ⟵ 0;
(2) loop: for k = 1; k ≤ 4; k + + do
(3)     for i = 1; i ≤ m; i + + do
(4)         for j = 1; j ≤ m; j + + do
(5)             if La_k i == Lb_k j then
(6)                 result + +;
(7)                 break loop;
(8)             end
(9)         end
(10)    end
(11) end
(12) if result == 4 then
(13)     return true;
(14) end
(15) else
(16)     return false
(17) end
```

Algorithm 2: Location matching.

versions $C([I_i]) = (I_i + P_i \times R_i) \bmod N_i$. In this way, it can make sure that even if only one attribute group of the user is the same as that of the recommended friends, the friends' personal information can still be decrypted by the key of this attribute group. Therefore, the users can know each other better.

4.2. Matching Algorithm. When receiving the user's ID and the information encrypted, the server runs the matching algorithm to find suitable recommended people for the user in the database. The matching algorithm mainly includes two parts: location matching and attribute matching.

4.2.1. Location Matching. The purpose of location matching is to find people whose district overlaps that of the user. As shown in Figure 6, the grey square represents $user_1's$ district, and the black square represents $user_2's$ district. If these two districts overlap, then it must look like one of the situations in Figure 3. The overlapping can appear in the upper left, upper right, lower left, and lower right in the square. To simplify the matching algorithm, we see the user's location as the center of a rectangular coordinate system and divide the district into four parts based on the quadrant of the rectangular coordinate system. It is shown in Figure 4. Each vertex in the subdistrict is evenly distributed. The basic idea of the proposed location matching algorithm is that we compare two users' districts. If each subdistrict has the same vertices with the corresponding one, then we consider these two users' districts overlap.(Because homomorphic encryption algorithm can only judge whether the information is equal, we adopt the local discretization method for the location matching. The more vertexes in the district are, the more precise the matching algorithm is, and vice versa.)

Algorithm 2 judges whether two users' districts overlap: *La* which includes four queues holds $user_a$'s district information and *Lb* holds $user_b$'s district information. The algorithm is going to check four queues with their corresponding one (Lines (2)). For each queue, all the vertexes will be compared with those in the corresponding one (Lines (3)-(4)). When there are vertexes matched, then the algorithm tags this queue as matched (Lines (5)-(8)). If all the queues are matched with their corresponding one, we consider that these two users' districts are matched and return true (Lines (12)-(14)). Otherwise, we return false (Lines (15)-(17)).

4.2.2. Attributes Matching. When the user makes a query, the server will receive query information which is in the form of cipher-text. We use $C([M])$ to represent it. Because the server has stored the information of other users, the matching algorithm will compare each record in the database with the query information to find the suitable recommended people. The operation result is computed according to the following algorithm, where $C([Ma_i])$ represents the i attribute group in the form of cipher-text of the $user_a$ and Q_i, N_i represent the i attribute group's parameters. Similarly, $C([Mb_i])$ represents the i attribute group of $user_b$ in the database in the form cipher-text. K represents the number of attribute groups. If $res_i = 0$, the i attribute group of $user_a$ matches that of $user_b$ in the database. Otherwise, they are mismatched, and the server will compare other user's attribute groups with the cipher-text of the user.

K: the number of attribute groups.
Input:
$C([Ma_i])$: the i attribute group of $user_a$ in the form of cipher-text($i = 1, 2, 3...k$);
$C([Mb_j])$: the i attributes group of $user_b$ in the form of cipher-text($i = 1, 2, 3...k$);
Q_i, N_i: the i attribute group's parameters set by the $user_a$;
Output: res: the matching result ($res = res_1, res_2....res_k$)
(1) **if** $Q_i \neq 0$ *AND* $N_i \neq 0$ **then**
(2) **for** $i = 1; i \leq K; i++$ **do**
(3) $res_i \longleftarrow ((C([Ma_i]) - C([Mb_j])) \times Q_i) \bmod N_i$;
(4) **end**
(5) **end**
(6) return res;

Algorithm 3: Attributes matching.

Algorithm 3 shows the attributes matching process. The explanation of the detailed analysis is as follows: We derive the calculation formula in the Algorithm 3. It can get the following form. Ma_i and Mb_i represent the i attributes groups of $user_a$ and $user_b$ in the form of plain-text separately. Ra_i represents the part of the key of the i attribute group of $user_a$ and Rb_i represents the part of the key of $user_b$.

$$\begin{aligned} res_i &= \left(\left(C\left([Ma_i]\right) - C\left(\left[Mb_j\right]\right)\right) \times Q_i\right) \bmod N_i \\ &= \left(\left(\left(C\left([Ma_i]\right) + P_i \times Ra_i\right)\right.\right. \\ &\quad \left.\left. - \left(C\left(\left[Mb_j\right]\right) + P_i \times Rb_i\right)\right) \times Q_i\right) \bmod N_i \\ &= \left(\left(C\left([Ma_i]\right) - C\left([Mb_i]\right)\right) \times Q_i + \left(Ra_i - Rb_i\right) \times P_i\right. \\ &\quad \left. \times Q_i\right) \bmod N_i \end{aligned} \tag{1}$$

Because $N_i = P_i \times Q_i$, then $(Ra_i - Rb_i) \times P_i \times Q_i \bmod N_i = 0$; if $Ma_i == Mb_i$, then $((C([Ma_i]) - C([Mb_i])) \times Q_i) \bmod N_i = 0$. Otherwise, $((C([Ma_i]) - C([Ma_i])) \times Q_i) \bmod N_i > 0$. Therefore, if $Ma_i == Mb_i$, then res=0; otherwise, $res > 0$.

Considering matching, the server could request that all of the attributes be matched, but generally we do not need this. Objectively, it is tough to find two people whose attributes are the same. Subjectively, the purpose of dating is not to find a mirror image of the user, but to find suitable friends. Therefore, when it comes to matching algorithm, we mainly consider that parts of the attributes are the same. The partial matching is divided into two parts: Strict Matching and Loose Matching. We will introduce them in detail.

(1) Strict Matching: When the user is sending his query, he must have specific demands looking for friends. The user needs to appoint some attribute groups. The Strict Matching Attributes algorithm should make sure that these attribute groups are matched. In this way, although the server saves a large number of attributes of users, only part of the attributes will be used in the matching algorithm. It not only meets the needs of the users but also saves the operation cost to a great extent, which makes the matching algorithm more efficient. For example, as shown in Figure 5, let one user's attributes be {age: 31, favorite sport: basketball, favorite movie: The Titanic}. Each attribute is a group. Then, he points out that the friend he wants should like the same movie and have the same age. When the server responds, the recommended friends' favorite sport could be different, but they must love to watch The Titanic and have the same age.

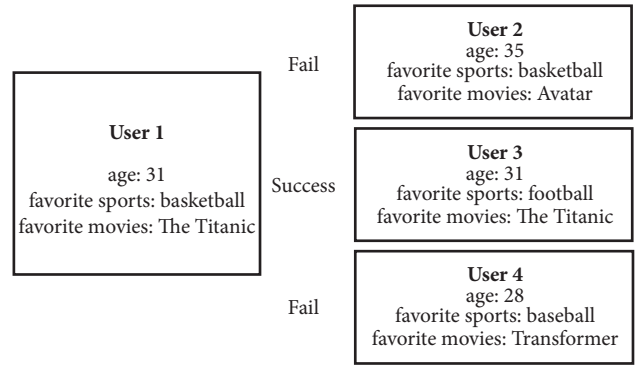


Figure 5: Strict Matching.

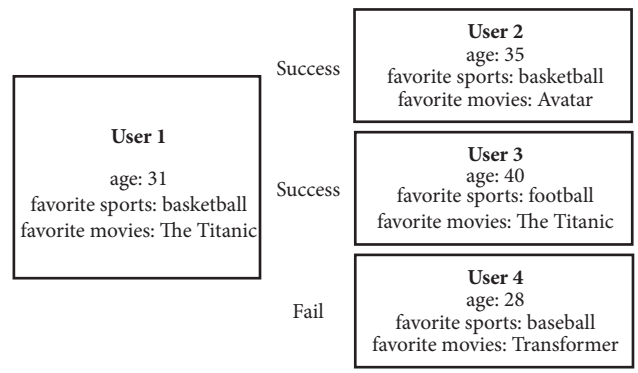


Figure 6: Loose Matching.

(2) Loose Matching: As for the Loose Matching algorithm, it only needs to make sure that part of the attribute groups which the user appoints is matched. If the user appoints N attributes and he gives a number M ($M \leq N$), the server only needs to check N attribute groups and makes sure whether there are matched M attribute groups. It does not need all the N attribute groups to be matched. It not only does not care whether all of the attribute groups are matched but also does not pay attention to which attribute groups appointed are matched. In other words, the Strict Matching is a particular case in Loose Matching.

Using the example above, shown in Figure 6, let one user's attributes be {age: 31, favorite sport: basketball, favorite movie: The Titanic}. Each attribute becomes a group. Then, he appoints movie and age as the matching condition and gives a number $M = 1$. When the server respondes, the recommended people's age and favorite movie could be different, but at least one of them is the same.

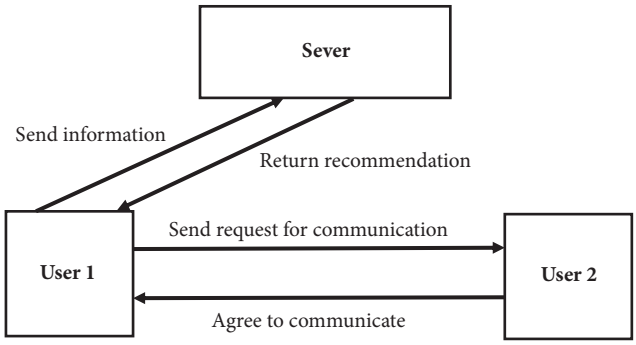


Figure 7: The decryption process.

4.3. Decryption. Having said that, we should divide the situations into different parts. First, it is not necessary to present the recommended friends' information to the users. In order to make sure that the user is not malicious dating, we can hand the right of showing the information to the recommended people. All of the information of any user who is searching friends is only used to run matching algorithm in the server. The information will not be sent to any user. The user could only get an "ID" of the recommended friends and an Internet connection which is used for chatting. Therefore, if the user sincerely wants to make friends, the server will tell him that this is the suitable person, but the next things should depend on him. In this circumstance, the decryption is useless. The process is shown in Figure 7.

Second, sometimes it is necessary to give users some information about the recommended people. This will help them to know each other and make friends more quickly. In this case, the decryption plays an important role. The user will receive the cipher-text packet, decrypt it using decryption algorithm, and get the plain-text. In some situations, the matching algorithm does not need all of the attributes to be matched.(Some attribute groups can be different.) Therefore, the decryption is only used for those attribute groups which are matched. The query user will send a request to the recommended people asking for information. If the recommended people agree on the query user's request, the query user can use the homomorphic decryption to obtain plain-text of the information. The decryption algorithm is shown in Algorithm 4.

In Algorithm 4, K is the number of attribute groups. M_i represents the i attribute group of recommended friends in the form of plain-text, and M represents all the attribute groups. P_i is the key of the i attribute group of the user. res is the matching result.

K: the number of attribute groups.
M_i: the i attribute group of recommended friends in the form of plain-text($i = 1, 2, 3...k$)
Input:
$C([M_i])$: the i attribute group of recommended friends in the form of cipher-text($i = 1, 2, 3...k$);
P_i: the i attribute group's key of user;
res: the matching result ($res = res_1, res_2....res_k$);
Output: M: all the attribute groups in the form of plain-text($M = M_1, M_2....M_k$);

```
(1) if P_i ≠ 0 then
(2)     for i = 1; i ≤ K; i++ do
(3)         if res_i == 0 then
(4)             M_i ⟵ C([M_i]) mod P_i
(5)         end
(6)     end
(7) end
(8) return M;
```

Algorithm 4: Decryption.

5. Defense for Specific Attack

Strictly, none of the encryption algorithms are safe mathematically. Certainly, the server or the hacker can still unremittingly try every possible key to decrypt all the information. Considering the cost of fee and time, it is unpractical in the current technology. However, they could act as a user to match people who have registered the same information in the server. In other words, as a hacker, he does not receive the whole information of the user, but some attributes. For example, if a hacker sends some attributes {age: 25, favorite sport: football, favorite movie: The Titanic} to the server. Then he points out that the friend he wants should have the same age and favorite sport. When the server respondes, there are some recommended friends sent back, and the hacker will know these people's age and favorite sport. In this situation, we adopt Obscure Matching based on dividing attributes into several groups, as well as district limitation, as we mentioned briefly in Section 4. In this section, we introduce our defense to this attack in detail.

5.1. Loose Matching. Dividing attributes into several groups makes room for partial matching, aiming to reduce the probability of information leakage. Based on the attribute groups, we propose Loose Matching further. It only requires that part of the attribute groups of the friends recommended by the server matches those appointed by the user. In this way, the attributes groups which are not matched by the user will not be known by the user. It not only meets the needs of the users but also reduces the probability of information leakage.

Using the example above, let the hacker's attributes be {age: 25, favorite sport: basketball, favorite movie: The Titanic}. Each attribute becomes a group. Then he points out that the recommended people should like the same sport and have the same age. When the server respondes, the recommended people may have the same age or like the same sport, but the hacker does not know which attribute is the same.

5.2. District Limitation. As we mentioned above, we talk about the location attribute. It also decreases the probability of information leakage. With the district limitation, the server can not recommend all the people who have the same attributes. In this way, we can not only reduce the burden

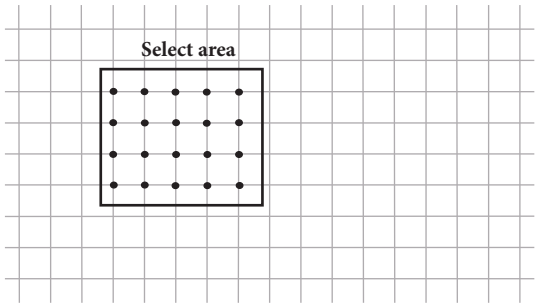


Figure 8: The district limitation.

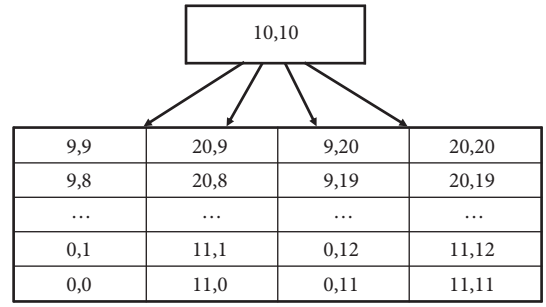


Figure 9: The location structure.

of the whole network but also prevent some malicious users to acquire other user's information. This also reduces the possibility for the hacker to pretend to be a user to acquire information. Below we give further analysis.

Suppose there are M hackers served by the server and there are K location vertexes covered by the database. The number of location vertexes near to the user is L. As shown in Figure 8, it is a select area. From the perspective of a user, without the district limitation, the possibility that the user is recommended to the hacker is 1. However, with the district limitation, the number of hackers near to the user is $(M \times L)/K$. Therefore, the possibility that the user is recommended to the hacker will be reduced to L/K. The more the location vertexes are, the smaller the possibility is.

For example, according to the longitude and latitude, there are 360×180 vertexes which are the integer coordinates in the world. Let us presume that there are 10000 hackers in the world. Based on the regulation of location attributes, only the people near to the user will be recommended. The number of the near coordinates which could be recommended to the user is 100. Therefore, with the district limitation, the possibility that the user is recommended to the hacker is $100/64800 = 0.0015$. The number of the hacker who can acquire the user's information is $(100 * 10000)/64800 = 1500$. It is clear that through the district limitation, the possibility of user information leakage has largely decreased. As for the users who are near to each other, we should believe that because they are close, they intend to make friends with each other. The information that they get is necessary.

6. Simulation

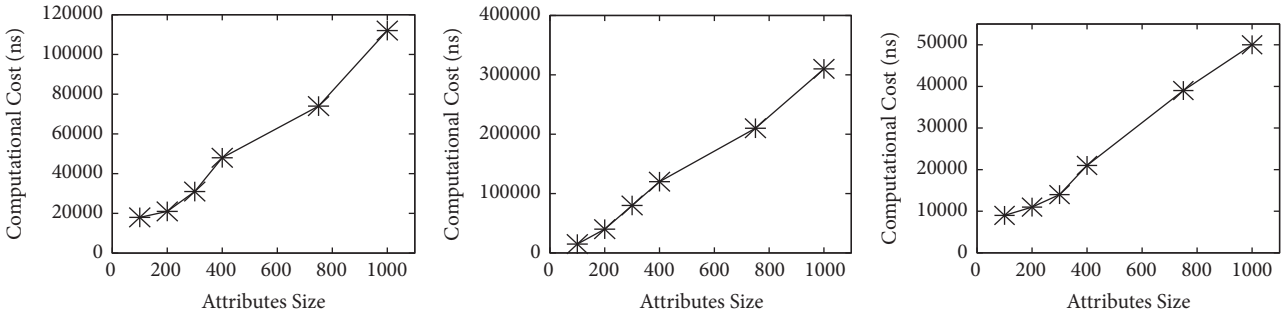
We build a project which is based on the C/S system to realize our framework. The client which resides in the user's machine is used to provide a service of recommending friends.

As mentioned before, the client mainly realizes dividing the attributes into groups and encrypting the user's information. When receiving recommended friends from the server, it also decrypts attributes which are matched. We code all of this with Java. As for user's information, the attributes are programmed directly as Class (Java). These are the core of the user's information which will be used frequently. Then, we build a pointer in the Class which points to the personal information. In this way, the runtime of the matching algorithm will be accelerated, and the storage space will be saved.

The function of the server is to run the location and attributes matching algorithms. For the attributes matching, we need to covert the texts into numbers before encryption and transform numbers into texts after receiving the recommended friends' information. Actually, these two conversions are the same essentially. Taking the conversion from numbers to texts, for example, we correspond each number to a letter according to the ASCLL. As for some letters corresponding to a number which only has two digits, we add 9 to the hundreds place. Hence, each letter corresponds to a three-digit number. In this way, each big number corresponds to a word. Let the big number '109117115105999' to be an example. The corresponding letters of each three-digit number are 109-m, 117-u, 115-s, 105-i, and 999-c. Therefore, the corresponding word of this big number is 'music'. For the location matching, as mentioned above, the server views the user's location as a center to build a district. Each district consists of four subdistricts. The coordinates of vertexes which compose different subdistricts are saved in different queues. When using the matching algorithm, we compare the four queues with their corresponding one separately. The detailed process of generating the district according to the user's coordinates is as follows:

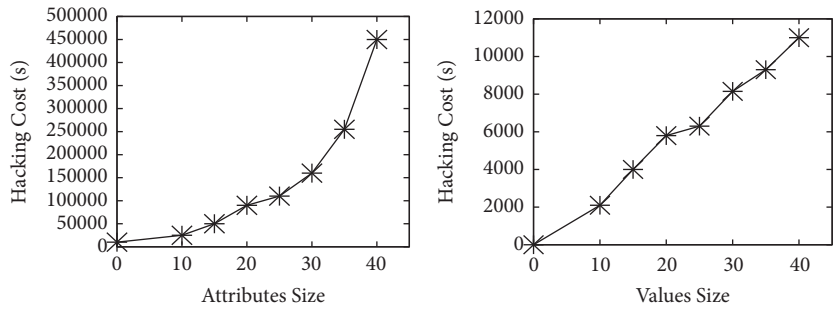
> We view the user's location as the center of the plane rectangular coordinate system. Take the first quadrant for example. We add from 1 to 10 to the longitude and the latitude of the user's current location to create 100 new vertexes to build a subdistrict. We do the same operation on the other quadrants.
>
> We use four queues to save the information of four quadrants. The vertex's coordinates from the same quadrants will be saved in the same queues. Figure 9 shows, when the user's coordinates are $(10, 10)$, how is the data of the subdistrict saved in the four queues.
>
> The client will encrypt all of the coordinates of vertex. At the same time, the order of the vertexes in each queue should be disorganized to make sure the server could not find the center vertex in the queues.

The server is also coded with Java. We built the server with many sockets, such as Register Socket, Match Socket, Modify Socket, etc. As the most crucial socket, the Match



(a) Computation cost of encryption (b) Computation cost of matching (c) Computation cost of decryption

Figure 10: The speed of encryption, matching, and decryption.



(a) Difference attribute size (b) Difference attribute values size

Figure 11: Possibility of decryption.

Socket receives the users' ID and attributes and sends the information in the server layers by layers to the database layer. Then the server will open the database, retrieve information, and run the matching algorithm [31]. After that, the server packs all of the suitable recommended friends' information and sends it back to the users. Finally, the client uses this information like users' ID to make a connection with the potential friends.

The experiments are run on two Windows machines with 2.4GHz CPU, 8 GB memory, 1TB 7200 RPM hard-disk. One is used for the server. The other runs the client.

In our simulation, we focus on several important metrics. The speeds of encryption, decryption, and matching as well as the possibility of decrypting the information by the server or the hacker.

6.1. The Speed of Encryption, Matching, and Decryption. Figure 10, respectively , illustrates how the time of encryption, matching, and decryption change with different attribute numbers M. Here, $M = 100, 200, 300, 500, 750, 1000$ and all the information is encrypted by the same two prime numbers, i.e., $P = 19961993$, $Q = 2013265921$. It is clear that the encryption, matching, and decryption costs are almost linearly proportional to M. As the the attribute size increases, the time needed to wait for sending information will be increased.

6.2. Possibility of Decryption. The possibility of the server decrypting the information is important since we need to make sure that, with the current technology, there is low chance to leak the information to the user. Therefore, we try to decrypt the cipher-text and record the cost from a hacker's angle. (We assume that there are hackers or the server does an inside job to act as a hacker.) We put M attributes in an attribute group, where M=10, 15, 20, 25, 30, 35, and 40, and each attribute has 5 values. For example, as for 30 attributes in one group, we use 300 big prime numbers divided into 150 groups to encrypt attribute groups. For each different combination of the values to one attribute group, we use different groups of prime numbers for decryption. If the server wants to decrypt the information, it will need to try 75 times on average (searching in disordered arrangement [32]). It is clear that the time of decryption by the server is almost squarely proportional to the numbers of attributes. As the number of attributes increases, the difficulty of the server decrypting the information will be increased.

Figure 11(a) shows the cost of hacking when the number of attribute values is invariant and the number of attributes changes. In this part, we test the hacking cost when the number of attribute values changes and the number of attributes is invalid. We take 10 attributes in an attribute group, and each attribute has 10, 15, 20, 25, 30, 35, and 40 values, respectively. For example, as for 30 values, we use 300 big prime numbers divided into 150 groups to encrypt attribute groups. For each different combination of the values

to one attribute group, we use different groups of prime numbers for encryption. If the server wants to decrypt the information, it needs to try 75 times on average (searching in disordered arrangement [32]). Figure 11(b) shows the time the server needs to decrypt the information. It is obvious that the time of decryption by the server is almost linearly proportional to the numbers of values of the attributes. As the number of values of attributes increases, the difficulty of the server decrypting the information will be increased. We can conclude that the chances of decryption by the server will be low if the number of attributes and the number of values of attributes are big enough.

7. Conclusion

In order to better protect the user's personal information on the friend-recommendation applications, we put forward the privacy protection mechanism based on zero knowledge. The client encrypts the information using homomorphic encryption and sends the cipher-text to the server. Then the server helps the user to find people who have the same attributes based on homomorphic encryption. In the whole process, the user does not need to worry about whether his information is in danger. Once his information leaves the client, it is in the form of cipher-text. Meanwhile, the server can still finish its work smoothly. We propose a method of district limitation for matching operation which can decrease the possibility of information leakage, and grouping attributes enhance the security of our framework. Finally, the defenses of grouping attributes and location limitation largely reduces the risk of being leaked by the hacker. Besides, the Loose Matching also provides much protection. This mechanism can realize that although a friend-recommendation system server does not know any users' information, it can help user match friends who are in accordance with the same attributes. This scheme not only has reference and practicality for practical application, but also has excellent scalability. It provides a strong basis for the future perfection and enhancement.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

An early version of the manuscript has been presented as conference paper in 2017 International Conference on Identification, Information and Knowledge in the Internet of Things [33].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (No. 61472044, No. 61772080, No. 61472403) and the Natural Science Foundation of Beijing (No. 4182062).

References

[1] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li, "Free market of crowdsourcing: Incentive mechanism design for mobile sensing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3190–3200, 2014.

[2] X. Zhang, Z. Yang, W. Sun et al., "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2016.

[3] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1-1, 2017.

[4] J. Wen-guang and S. Yu-qing, "Personalized privacy protection for third-party service platform," *Journal of Lanzhou University, Natural Sciences*, vol. 48, no. 4, pp. 85–90, 2012.

[5] D. Chen and H. Zhao, "Data security and privacy protection issues in cloud computing," in *Proceedings of the Proceeding of the International Conference on Computer Science and Electronics Engineering (ICCSEE '12)*, vol. 1, pp. 647–651, Hangzhou, China, March 2012.

[6] A. Vapen, N. Carlsson, A. Mahanti, and N. Shahmehri, "Information sharing and user privacy in the third-party identity management landscape," *IFIP Advances in Information and Communication Technology*, vol. 455, pp. 174–188, 2015.

[7] K. Vu, R. Zheng, and J. Gao, "Efficient algorithms for K-anonymous location privacy in participatory sensing," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 2399–2407, Orlando, Fla, USA, March 2012.

[8] L. Sweeney, "*k*-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[9] F. Yu, M. Chen, B. Yu, W. Li, L. Ma, and H. Gao, *Privacy Preservation Based on Clustering Perturbation Algorithm for Social Network*, Springer Science and Business Media, 2017.

[10] C. Zhipeng and Z. Xu, *A Private and Efficient Mechanism for Data Uploading in Smart Cyber-Physical Systems*, Transactions on Network Science and Engineering (TNSE).

[11] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[12] L. Guo, C. Zhang, and Y. Fang, "A Trust-Based Privacy-Preserving Friend Recommendation Scheme for Online Social Networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 413–427, 2015.

[13] B. K. Samanthula, L. Cen, W. Jiang, and L. Si, "Privacy-preserving and efficient friend recommendation in online social networks," *Transactions on Data Privacy*, vol. 8, no. 2, pp. 141–171, 2015.

[14] 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.

[15] X. Zheng, Z. Cai, G. Luo, L. Tian, and X. Bai, "Privacy-preserved community discovery in online social networks," *Future Generation Computer Systems*, 2018.

[16] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *Proceedings of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, May 2017.

[17] L. Zhang, Z. Cai, and X. Wang, "FakeMask: A Novel Privacy Preserving Approach for Smartphones," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 335–348, 2016.

[18] Z. He, Z. Cai, and X. Wang, "Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks," in *Proceedings of the 35th IEEE International Conference on Distributed Computing Systems (ICDCS '15)*, pp. 205–214, July 2015.

[19] X. Qin, Y. Luo, N. Tang, and G. Li, "DeepEye: An automatic big data visualization framework," *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 75–82, 2018.

[20] L. Shi, Y. Wu, L. Liu, X. Sun, and L. Jiang, "Event detection and identification of influential spreaders in social media data streams," *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 34–46, 2018.

[21] H. Zang and J. Bolot, "Anonymization of location data does not work: a large-scale measurement study," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom '11)*, pp. 145–156, ACM, September 2011.

[22] R. Lu, X. Lin, Z. Shi, and J. Shao, "PLAM: A privacy-preserving framework for local-area mobile social networks," in *Proceedings of the 33rd IEEE Conference on Computer Communications, IEEE INFOCOM 2014*, pp. 763–771, Canada, May 2014.

[23] A. C. Squicciarini, F. Paci, and S. Sundareswaran, "PriMa: A comprehensive approach to privacy protection in social network sites," *Annals of Telecommunications-Annales des Télécommunications*, vol. 69, no. 1-2, pp. 21–36, 2014.

[24] 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*, 2016.

[25] H. Li, H. Zhu, S. Du, X. Liang, and X. Shen, "Privacy leakage of location sharing in mobile social networks: attacks and defense," *IEEE Transactions on Dependable and Secure Computing*, no. 99, 2016.

[26] J. Yu, B. Zhang, Z. Kuang, D. Lin, and J. Fan, "IPrivacy: image privacy protection by identifying sensitive objects via deep multi-task learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1005–1016, 2017.

[27] P. Ilia, I. Polakis, E. Athanasopoulos, F. Maggi, and S. Ioannidis, "Face/off: preventing privacy leakage from photos in social networks," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, (CCS '15)*, pp. 781–792, October 2015.

[28] Z. Khan, "Qbuasi-Linear Time Fully Homomorphic Public Key Encryption Algorithm (ZK111)," *Journal of Theoretical Physics and Cryptography*, vol. 1, no. 1, pp. 14–17, 2012.

[29] Y. Yang, S. Zhang, J. Yang, J. Li, and Z. Li, "Targeted fully homomorphic encryption based on a double decryption algorithm for polynomials," *Tsinghua Science and Technology*, vol. 19, no. 5, pp. 478–485, 2014.

[30] K. Hariss, H. Noura, and A. E. Samhat, "Fully Enhanced Homomorphic Encryption algorithm of MORE approach for real world applications," *Journal of Information Security and Applications*, vol. 34, pp. 233–242, 2017.

[31] N. Ogura, G. Yamamoto, T. Kobayashi, and S. Uchiyama, "An improvement of key generation algorithm for Gentry's homomorphic encryption scheme from ideal lattices," *Journal of Math-for-Industry (JMI)*, vol. 3B, pp. 99–106, 2011.

[32] M. Li, J. Li, and C. Huang, *A Credible Cloud Storage platform based on Homomorphic Encryption*, Beijing University of Posts and Telecommunication, 2012.

[33] W. Wang and S. Wang, "Privacy Preservation for Dating Applications," *Procedia Computer Science*, vol. 129, pp. 263–269, 2018.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 3530123, 18 pages
https://doi.org/10.1155/2018/3530123

Research Article

Differentially Private Recommendation System Based on Community Detection in Social Network Applications

Gesu Li,[1] Zhipeng Cai ,[1,2] Guisheng Yin,[1] Zaobo He,[3] and Madhuri Siddula[2]

[1]*College of Computer Science and Technology, Harbin Engineering University, Heilongjiang, China*
[2]*Department of Computer Science, Georgia State University, Georgia, USA*
[3]*Department of Computer Science and Software Engineering, Miami University, Ohio, USA*

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

Received 17 June 2018; Revised 28 August 2018; Accepted 5 September 2018; Published 3 October 2018

Guest Editor: Liran Ma

The recommender system is mainly used in the e-commerce platform. With the development of the Internet, social networks and e-commerce networks have broken each other's boundaries. Users also post information about their favorite movies or books on social networks. With the enhancement of people's privacy awareness, the personal information of many users released publicly is limited. In the absence of items rating and knowing some user information, we propose a novel recommendation method. This method provides a list of recommendations for target attributes based on community detection and known user attributes and links. Considering the recommendation list and published user information that may be exploited by the attacker to infer other sensitive information of users and threaten users' privacy, we propose the CDAI (Infer Attributes based on Community Detection) method, which finds a balance between utility and privacy and provides users with safer recommendations.

1. Introduction

The recommender system is one of the most popular technologies that is used in any e-commerce websites. This system is mainly developed for the ease of user interaction with the website and to suggest more products to increase the business. Many e-commerce websites such as Amazon, Netflix, and Movielens recommend products based on users' preferences (or interest). The recommender system suggests new products based on various factors including user preferences, item features, user purchase history, and other additional information such as time and space data. By introducing the recommendation system, we solve the search time of a user that in turn reduces the complex and dynamic data processing at the server. However, with the invention of social networking, recommender system has taken a drastic shift from recommending products based on user purchase history to user social activity. Most of social networking websites like Facebook, Google+, and Twitter use recommender system for pop up advertisements that might cater to the users of different tastes. Also, social networking websites like Facebook recommend communities that the user might be interested in based on the profile that he has created while signing up.

The current use of recommender system in online social networks is a content-based or a hybrid system that utilizes collaborative filtering method. This method recommends products based on user's community preferences such as user's ratings and purchase history but ignores the user and item attributes. Content-based filtering or information filtering methods [1, 2] typically match query words or other user data with item attribute information, ignoring data from other users. The hybrid recommender system combines the above two methods but ignores the user's link. As the awareness of user's privacy has increased, most users in online social networks hide personal information. Users tend to opt for selective data publication while registering in a social network. For example, some users might cancel location-based services or do not fill out any personal information. This tendency creates severe lack of information and challenges for the service provider's recommendation service. Additionally, the models mentioned above also have some limitations in dealing with data sparsity and cold start-up problems [3–6].

Although there is a lack of information due to privacy settings, users still disclose at least some of their private information, for example, rating a movie or sharing "like" or "dislike" for a photo. Some users might also reveal sensitive information and hence become vulnerable to the attacker who utilizes background knowledge for retrieving information. These knowledge-based attacks also infer personal identity information. For example, Lewis et al. found correlations of provided favored books, movies, and music [7]. In this paper, we focus on users' inherent data privacy. We assume that an attacker can anonymously collect user's data from a social network. Some users reveal their sensitive information, while others, as privacy-conscious users, have some recommended data privacy in their long-term use. However, attackers can use various methodologies to further infer user's sensitive information. Therefore, it is the main idea of this paper to add noise to the released data to hide sensitive information and to protect users' privacy.

This paper uses the real world data of online social networking websites like Facebook and Google+ for the experiments. Also, we focus on community detection as our output for the recommender system. People, as social animals, like to find groups of people who have something in common with themselves. With the advent of social networks, it has become more accessible for people to connect to those groups of users who share common interests, even if there is no friendship between them in the real world. For example, there are various movie communities in Google+. Users can join any group based on their preferences although there is no link to the group before. Based on this idea, we propose the CDAI (Infer Attributes based on Community Detection) method to provide a list of recommendations for users in different communities. At the same time, a privacy protection strategy is proposed based on the differential privacy to balance the utility and privacy.

In this work, we focus on two main issues: (1) When the user is missing much information, how to provide the recommendation list with high accuracy; (2) How to protect users' inherent data privacy while publishing data. Following is the summary of our contributions and improvements over the previous works:

(i) Two definitions are proposed, one is to define the user, and the other is to classify the attributes according to the privacy type. In this paper, we divide users into two categories, positive and negative (see Definition 8). Additionally, this paper divides the user attributes into two categories, inherent attributes and recommendation attributes (see Definition 9).

(ii) The recommendation framework of the CDAI method is proposed (see Section 5). We have built a recommendation system using community discovery and attribute dependency. The accuracy of the proposed method is superior to the accuracy of the recommender system based on user attributes and connections.

(iii) Based on the differential privacy theory and its properties, we propose the privacy-utility strategy, using differential privacy technology to protect users' sensitive information. Additionally, we also propose methods that prevent attackers from utilizing the results of CDAI to reversely infer privacy.

The remainder of the paper is organized into eight sections as follows. Section 2 provides the related work which consists of three ideologies used in this paper, including community detection, inference attack, and privacy protection. Section 3 talks about preliminaries for community detection. Section 4 defines the CDAI method, including social network model, community, dependency relationship and privacy-utility, input and output, and task definition. We propose our algorithm and the description of inference attribute and privacy protection in Section 5. Sections 6 and 7 are the evaluation and conclusion, respectively.

2. Related Works

2.1. Recommender System. The recommender system is mainly used to evaluate and predict users' preferences for the project. Based on the output type, the recommender system can be divided into the following three categories: rating prediction, ranking prediction (top-n recommendation), and classification. The goal of the first category is to fill the missing user-project scoring matrix. The goal of the second category is to provide a ranking of n items for users. The final category, classification, is to classify the candidates into the correct recommendation category. Social networking platforms vastly differ from e-commerce platforms. The e-commerce platform provides a reference standard for users to purchase a project through the evaluation of the project. However, social networks are more about providing users with "novelty" and finding things they might be interested in. In social network sites such as Facebook, if users' friends have watched a movie, they can share it on their page as well as their friends. Users who see this share on their page serve as a personal advertisement for the movie. So a user's evaluation of a project is only "watched" or "not watched", "like" or "dislike", instead of rating every item. Based on the characteristics of the social network platform itself, this paper selects the last type of task output.

The current recommender system based on the social network is broadly based on the framework of matrix factorization [8] and probabilistic matrix factorization [9]. STE [10] believes that ratings are generated by users and their trusted friends' interest. Sorec [11] uses the user preference vector to decompose the link matrix. TrustMF [12] constructs a link from a specific feature vector of trustees and then integrates it into a user preference vector to predict missing ratings. First of all, most of the above recommender systems are still based on e-commerce platform project recommendation. The results are based on user ratings of the project. However, this recommendation method has serious limitations in cold start and data sparsity; secondly, the recommendation research based on the social network mostly focuses on the recommendation of friends' link relationship. However, with the development of online networks, social networks and e-commerce are gradually breaking down each other's boundaries. The researcher can predict the target item by combining the user's link relationship with their attributes

and item attributes. This paper presents a novel recommendation method, to use community detection and user attributes to provide recommendations to the users. Therefore, it can effectively reduce the inconvenience caused by data sparsity.

2.2. Inference Attack. In the study of attribute inference attack, the target of the attacker is to spread the attribute information of social network users to users with incomplete attribute data. The attacker could be any party (e.g., online social network service, cyber criminal, advertiser, and so on). These attackers may be more interested in users' privacy attributes. They attack users' privacy by collecting public data. In addition to the risk of privacy leaks, it is also possible for an attacker to perform various security sensitive activities using inferred user attributes, such as spear phishing [13] and attacking personal information based backup authentication [14]. Reference [15] proved that an attacker (the provider of the recommender system) can use the machine learning classifier to predict the gender of the user based on the user's movie rating data. In social networks (e.g., Facebook, Twitter, and so on), public data on users include lists of users' preferences (such as movies they like or share) and lists of users' friends. Some researchers [16–23] proved that the attackers use machine learning classifiers to infer target users privacy information based on user's published data.

The current attribute inference attacks are mainly divided into two types, friend-based and behavior-based. Friend-based attacks [18, 19, 24–29] are mainly based on the information that users and friends have disclosed as well as the information of social relationship structure to infer users' sensitive attributes to attack them. Behavior-based attack [15, 17, 30, 31] is based on the public attribute information of the user, to find users with similar attributes, culture, and hobbies. By using this public information, the user's behavior attributes are inferred. In Section 3, the definition of an attacker is provided. By using the prediction results of this paper and combining with the background knowledge such as the user's public information, the attacker could deduce the sensitive attributes (such as location, political view, or sexual orientation) of the user in reverse and thus poses a threat to users' privacy.

2.3. Privacy Protect. At present, the privacy protection researches mainly focus on anonymity and access control. Anonymity is one of the most important methods for protecting privacy in the social network. There are two possible anonymization methods: node anonymity and edge anonymity. The former method hides the node information, while the latter prevents attacker based on the known information and relationship to infer node's real identity through adding or deleting edges [32]. The most popular method is to combine the two methods to hide node information and get the better privacy.

Sweeney [33] proposed the k-anonymity method. This method does not consider sensitive attributes and hence is prone to attacks like homogenous and background knowledge. Hey et al. [34] proposed K-candidate anonymity method based on Sweeney. They aggregate nodes to a different partition and published the number of partitioned nodes and edge degree between inter and outer of partition, using this anonymity graph to study the original feature.

Edge anonymization can be done by randomly removing a certain number of edges and adding the same number of random edges. Vuokko and Terzi [35] studied the reconstruction mechanism of social networks. In this mechanism, the authors have randomized both the structure and the attributes. However, they deemed that the reconstruction can be finished in polynomial time.

Differential privacy is a privacy definition proposed by Dwork in 2006 for the privacy disclosure of statistical databases [36]. They defined the computing result of the data set to be nonsensitive for a single record. Difference privacy can solve two defects in traditional privacy protection model. (1) It does not consider any background knowledge which the attacker has. (2) It has the strict definition and provided a quantitative assessment method. Many scholars begun the research in difference privacy field. For example, McSherry [37] proposed a privacy integration query mechanism (PINQ). PINQ proposed an algebraic method to describe the protection of privacy data analysis, and they ensured that the results satisfied difference privacy. Differential privacy is also frequently studied on the context of index [38–46]. This paper combines matrix manipulation of attributes with differential privacy technology, and we propose a novel privacy protection method to prevent attackers to infer user's sensitive information.

3. Preliminaries

3.1. Community Detection. Community Detection is a study in which nodes are divided into communities. There are various ways in which the nodes can be divided into communities and the splitting method is called "cluster" method. Broadly, all the clustering methods can be divided into overlapping community methods and nonoverlapping community methods. The community detection of this paper is based on Louvian algorithm [47]. We assume that users are only in a community. Because community detection is divided according to the user's target attribute. The target attribute in this paper belongs to recommendation attribute and has multiple values. Therefore, we utilize a method called "CDAI" which is a overlapping community method.

The strength and weakness of intimacy in the community, the quality of division, require a measure. Moreover, that standard is modularity. Newman first proposed the concept of modularity in 2004 [48]. Newman also proposed a large-scale community division approach. However, this approach works very slowly with big data. Therefore, many scholars proposed many improved algorithms based on Newman's research, and Louvian was one of them. Louvian algorithm is proposed based on modularity, and it has been improved so that high accuracy results can be obtained quickly under large-scale community division. At the same time, [47] also proposed the concept of modularity gain as a standard to classify nodes into the community. The formula is as follows.

$$Q = \frac{1}{2b}\sum_{i,j}\left[W_{i,j} - \frac{k_i k_j}{2b}\right]\delta\left(c_i, c_j\right) \quad (1)$$

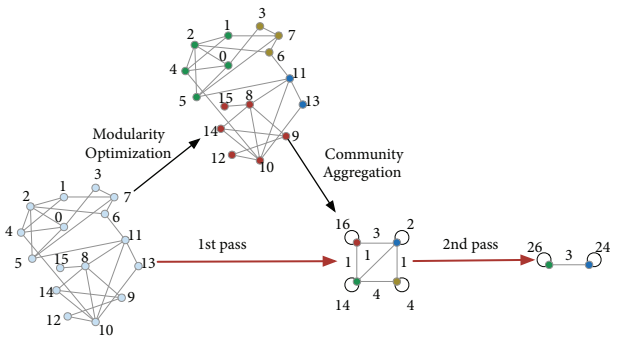


Figure 1: Visualization of the steps of the algorithm.

where $W_{i,j}$ represents the weight between node i and node j. $k_i = \sum_j W_{i,j}$ is the sum of the weights of the edges attached from vertex i, and c_i is the community to which vertex i is assigned. The δ-function $\delta(c_i, c_j)$ decides whether to put i and j in the same community. δ is 1 if u=v and 0 otherwise and $b = (1/2)\sum_{i,j} W_{i,j}$. Simplify formula (1) to formula (2):

$$Q = \sum_c \left[\frac{\sum_{in}}{2b} - \left(\frac{\sum_{tot}}{2b} \right)^2 \right] \tag{2}$$

where $\sum_{in}$ is the sum of the weights of the links inside C and $\sum_{tot}$ is the sum of the weights of the links among nodes in C.

The gain in modularity is ΔQ. If the node itself is a community, compute the gain when the node moved into other communities.

$$\begin{aligned} \Delta Q &= \left[\frac{\sum_{in} + k_{i,in}}{2b} - \left(\frac{\sum_{tot} + k_i}{2b} \right)^2 \right] \\ &\quad - \left[\frac{\sum_{in}}{2b} - \left(\frac{\sum_{tot}}{2b} \right)^2 - \left(\frac{k_i}{2b} \right)^2 \right] \\ &= \frac{1}{2b} \left(k_{i,in} - \frac{\sum_{tot} k_i}{b} \right) \end{aligned} \tag{3}$$

where k_i is the sum of the weights of the links incident to node i, $k_{i,in}$ is the sum of the weights of the links from i to nodes in C and b is the sum of the weights of all the links in the network. A similar expression is used in order to evaluate the change of modularity when i is removed from its community. In practice, one therefore evaluates the change of modularity by removing i from its community and then by moving it into a neighbouring community. Visualization of the steps of the algorithm is as follows.

In Figure 1, each part is made of two phases: one where modularity is optimized by allowing only local changes of communities and one where the communities found are aggregated in order to build a new network of communities. The passes are repeated iteratively until no increase of modularity is possible.

3.2. Differential Privacy. Due to the particular attack hypothesis and specific background knowledge, the privacy protection model based on anonymity fails to carry out quantitative analysis on the privacy protection intensity, so it has significant limitations in practical application. However, the differential privacy model makes it impossible for an attacker to identify whether a record is in the original data table, no matter what background knowledge he has. The formal definition of differential privacy is as follows.

Definition 1 (ϵ-differential privacy). A randomized algorithm M satisfies ϵ-differential privacy, if for any two datasets D and D' that differ only in one record, and for any possible output O of M, we have

$$P[M(D) = O] \leq \exp^{\epsilon} \times P\left[M\left(D'\right) = O\right] \tag{4}$$

where the probability of an event is denoted by $P[\cdot]$.

Definition 2 (Laplace mechanism sensitivity). For dataset D, given the function f: $D \longrightarrow R^d$. If the sensitivity of the function is Δf, then the random algorithm is as follows.

$$M(D) = f(D) + X \tag{5}$$

$X \sim Lap(\Delta f/\epsilon)$ is a random noise and follows the Laplace distribution of scale parameter $\Delta f/\epsilon$.

There are many methods of achieving differential privacy, and the most widely used are the Laplace mechanism [36] and the exponential mechanism [49]. Laplace mechanism is only applicable to numerical query results. The index mechanism is applied to query results by type. The exponential mechanism uses random sampling to satisfy specific distribution to realize differential privacy instead of adding noise. The central principle of the exponential mechanism is to define a practical evaluation function q and calculate a practical value for each output scheme. The output scheme with high score is more likely to be published, to ensure the quality of published data. The selection of the evaluation function q must have the lowest sensitivity possible. Sensitivity is the maximum change to the query result caused by deleting one arbitrary tuple in the dataset. It is a crucial parameter to add noise. Its specific definition and formula are as follows.

Definition 3 (exponential mechanism sensitivity). Given a practical evaluation function q, the sensitivity of q is defined as

$$S(q) = \max_{D_1, D_2, r} \|q(D_1, r) - q(D_2, r)\| \tag{6}$$

where D_1 and D_2 are datasets with only one record difference between any pair, and r represents any legitimate output.

According to the definition of sensitivity, Theorem 4 can be obtained.

Theorem 4. *Given dataset D, q is a utility valuation function for all output of dataset D. For dataset D and function q, if algorithm K satisfies the probability of output r and is proportional to* $\exp(\epsilon q(D, r)/2S(q))$, *then algorithm K satisfies* ϵ*-differential privacy.*

The ϵ-differential privacy model has the following properties.

Property 5 (sequence composition property). Assuming random algorithm combination $M = M_1, M_2, \ldots, M_n$. For the same dataset D, M_i provides ϵ_i-differential privacy, and M provides $\sum_{i=1}^{n} \epsilon_i$-differential privacy.

Property 6 (parallel composition property). Assuming random algorithm combination $M = M_1, M_2, \ldots, M_n$. For the disjoint subset of dataset D, ϵ-difference privacy is satisfied, respectively. Then M provides maximum ϵ-difference privacy protection for D.

4. Problem Statement

4.1. Social Network Model

Definition 7 (social network). A social network is an undigraph represented by $G(V, E, \chi)$. It consists of user set, friendship links, and the set of user attributes. The user set includes positive users and negative users represented by V_P and V_N, respectively, and $u_i, u_j \in V(1 \leq i, j \leq |V|)$. Friendship link is represented by E; in this paper all links are unweighted undigraph. But when dealing with similarity (Definition 11), links become weighted undigraph. The set of user attributes is denoted by χ, which consists of recommended attributes and inherent attributes represented by χ_R and χ_I, $(\chi_R, \chi_I) \subseteq \chi$ $\chi = \chi_R \cup \chi_I$. For an arbitrary user $(u_i, u_j) \in V(1 \leq i, j \leq |V|)$, their friendship links $(e_i, e_j) \in E$ also indicate $(e_j, e_i) \in E$.

Definition 8 (positive users and negative users). We divided users into two categories: positive users and negative users. We defined positive users as the people who filled most profile, nonsensitive for information, like sharing the information with other people. Positive users are denoted by V_P. V_N represented negative users, who have strict privacy awareness, blanking much information, inactivity on the online social network. These negative users are our target users. $u_i, u_j \in V(1 \leq i, j \leq |V|)$, $V_P \cup V_N = V$. In this paper, we consider the published information including the friend links and users attributes.

Definition 9 (attribute set). In this paper, we assume that the recommendation attributes are the items in the traditional recommender system, and the attribute value is the specific name of the items. This paper uses known recommendation attributes and some inherent attributes to predict user preferences and provide users with a list of recommendations for relevant content. The target attribute in this paper is a recommendation attribute, and the community is divided according to the target attribute. The inherent attribute is defined as the user's sensitive privacy information, which includes the user's age, gender, political inclination, address, and other personal data. All the inherent information of the user is regarded as an inherent attribute. Generally, such an attribute is a single value attribute. The recommendation attribute is considered as nonsensitive privacy information, which includes books, games, movies, and other information that users like. All items that provide services to users and can be recommended to users can be called recommendation attribute, which usually has multiple attribute values. For the convenience of the experiment, all single value attributes are regarded as inherent attributes in this paper, and all multivalue attributes are considered as recommendation attributes.

For any user u_i, $u_i \in V(1 \leq i \leq |V|))$, attribute set is denoted by $\overrightarrow{X_i} \subset \chi(1 \leq i \leq |V|)$, $x_j \in \overrightarrow{X_i}(1 \leq j \leq |\overrightarrow{X_i}|)$. Attribute category is T, $T_m \subset T(1 \leq m \leq |T|)$. T is the set of attribute categories in the social network, including the inherent category T_I and recommended categories T_R, $(T_I, T_R) \subseteq T$. For a certain attribute category, L is used to represent the attributes value also viewed as the label. $L_m \subset T_m$, $l_n \in L_m(1 \leq n \leq |L_m|)$.

For example, $x_i = \{T_{I_m} : l_1; \ldots; l_n\}$, which means x_i is for category T_{I_m} with value list $l_1; \ldots; l_n$ $n \geq 1$.

Different attribute categories have a differential number of attributes values, and the value can be single or multiple. Attributes such as gender and age need just single value. For categories like "Favorite movies", the input can be "Titanic" and "Darkest Hours". This kind of category has multiple values. Moreover, categories may be with none value for some users, for example, "Religion view". In specific applications, a user can determine which categories are sensitive and hide the categories. For example, Facebook users can directly hide their sensitive attributes in their profile. Following is an example:

$G = (V, E, \chi)$

$V = \{u_1 = Tom, u_2 = Ann\}$

$\chi = \{X_1, X_2\}$

T = {Favorite Movies, Favorite Books, Age, Gender, Political View}

T_R = {Favorite Movies, Favorite Books}

T_I = {Age, Gender, Political View}

$\overrightarrow{X_1}$ = {x_1 = { Favorite Movies: Avatar, Iron Man}; x_2 = {Favorite books: Machine Learning, AI }; x_3 = {Age: 26}; x_4= { Gender: Male }; { Political View: The Liberal Party}}

$\overrightarrow{X_2}$= {x_1 = {Favorite Movies: Iron Man, Bat Man}; x_2 = {Favorite Books: The Moon and Sixpence}}

$e_{1,2} \in E$ $e_{2,1} \in EC$ have five categories: two of them are recommendation attribute categories T_R and the others are inherent attribute categories represented by T_I; two users are u_1 and u_2. In this case, u_1 published two favorite movies, two favorite books, age, gender, and political view. However, u_2 released two favorite movies, one favorite book. From the case we can infer u_1 is a positive user, and u_2 belongs to negative user. We know the relationship between u_1 and u_2; they are friends.

4.2. Community Detection

Definition 10 (community). The community collects the nodes which have the same feature in the network. The vertices within the community are tightly linked, a low density between-group edges [50]. With the growth of the network scale, the community scale becomes bigger, and so is the number of nodes. We merge the original relationship between users and attributes by dividing the community while building a strong link in the community. Users having

same attributes in a community establish a new graph represented by G'.

In this part, to obtain the similarity of attributes and relationship links, getting a new social network graph G', where u_i and u_j have link, return to 1, otherwise return to 0. At the same time, u_i and u_j have a lot of same attributes, which proved they have tight relationship. The similarity is equivalent to the weight of user edges.

$$G' = G \cap SimA = \begin{cases} 0, & e_i \cap e_j = 0 \ or \ SimA = 0 \\ SimA, & e_i \cap e_j = 1 \ and \ SimA \neq 0 \end{cases} \quad (7)$$

Definition 11 (similarity of attributes between users). To obtain the similarity between users, use Jaccard index to measure that. It is defined as the size of the intersection divided by the size of the union of the sample sets.

$$SimA(i, j) = \frac{\left| x_i \cap x_j \right|}{|X|} \quad (8)$$

where x_i and x_j represented the attributes of u_i and u_j and the set of all attributes is denoted by X.

4.3. Attribute Dependence

Definition 12 (the accuracy of dependence). $\vartheta_{m \in M}^{T_R}(G)$ is the accuracy of dependence, classifiers set is M, and m is the specific classifiers in the set. $m_n \in M \ (1 \leq n \leq |M|)$, where n is the number of classifiers. T_R represents the target recommended attribute type.

Definition 13 (highest-dependence). Experiment used the selected classifiers to train. Remove an attribute category in each train, and find the attribute category that causes the maximum gain change, which is the highest-dependence for the target attribute category. If there are several attributes that result in large change, then regard them as highest-dependence.

$$\max \Delta_{m \in M}^{T_R} = \vartheta_{m \in M}^{T_R}(G)\Big|_T - \vartheta_{m \in M}^{T_R}\left(G'\right)\Big|_{T' = T - t_i} \quad (9)$$

When removing an attribute category, the gain of accuracy of recommended attribute category is denoted by $\Delta_{m \in M}^{T_R}$. T' is the set of rest attribute categories, and these attribute categories are the removed low-dependence attribute categories t_i.

4.4. Attacker and Attack Model. In general, an attacker can be anyone who is interested in user attributes. However, the attackers in this paper mainly refer to those cyber criminals who use public background knowledge to infer user attributes and attack users. Cyber criminal uses user attributes to perform certain cyber attacks, for example, spear phishing attacks [13] and personal information attacks based on backup authentication [14]. These behaviors will leak users' privacy and may cause users to lose property or even threaten their security.

In this paper, we assume that data is published from the perspective of social network providers and cyber criminal has some background knowledge. The knowledge of cyber-criminal is $K(V^K, E^K, \chi^K)$, where $V^K = V, E^K = E, \chi^K \subseteq \chi$. χ^K is the set of attributes obtained by machine learning, which is strongly dependent on the target attribute. Publish the set of extracted attributes. Positive users post complete information whereas negative users belong to the user group with strong privacy awareness, and they only release a small amount of information or no information to the outside world. Therefore, the goal of this step is to reduce the exposure of user information. Publish the attributes set after dimension reduction. Hide irrelevant or weakly related attributes to protect some user information.

4.5. Utility Based on Privacy Protection. This paper aims to propose a method to provide more services for negative users and, meanwhile, balance the utility-privacy tradeoff. The existing definition of privacy has difference privacy, K-anonymity [33] and L-diversity [51], which are only for inherent data and are not suitable for inferring attribute. In the current research work, the usual method is adding noise in original data. Rather than the traditional method, we infer attributes for negative users whose data are incomplete. However, consider the attackers use background knowledge to inference attack. Therefore, data-sanitization is necessary, which could protect user privacy, but over data-sanitization will lead to a reduced utility. Based on the above problem, utility is based on privacy protection as follows.

Definition 14 (utility). Given the social network G, stronger protection needs more noise, which leads to less utility. Accordingly, there is the interaction of constraints between privacy and utility. When utility satisfied the condition, get the maximum utility under the good privacy protection. The condition is given as follows.

$$\zeta \geq \frac{\Delta_{m \in M}^{T_R}}{\vartheta_{m \in M}^{T_R}(G)} \quad (10)$$

where ζ approximates to 1, which means utility better. Otherwise, it is close to 0, utility lower.

4.6. Input and Task. Based on the above definitions, given the input and output definitions about this paper. The user-specified thresholds on privacy-utility are given in Section 7.

Input. Social graph is denoted by $G(V, E, \chi, T, L)$, where user set V includes V_P and V_N. Friendship link set is denoted by E, the set of user attributes is denoted by χ, and the set of attribute categories T, $T = T_R \cup T_I$. The set of labels L, $L = L^K \cup L^U$.

The set of known labels for users $u_i \in V^K$ is L^K, where V^K is the set of users with known labels. V^K includes known V_P and V_N. L^U is the set of unknown labels for users $u_i \in V^U$, where V^U is the set of users with unknown labels, mainly the negative users. User-specified utility threshold is denoted by ζ.

Table 1: Description of symbols.

Symbol	Description
Q	The modularity, evaluation index of community detection
b	$b = (1/2)\sum_{i,j} W_{i,j}$, b is the sum of the weights of all the links in the network
$W_{i,j}$	The weight between node i and node j
k_i	The sum of the weights of the edges attached from vertex i
$k_{i,in}$	The sum of the weights of the links from i to nodes in C
C	The community. $c_i \in C$
M	An algorithm. $m \in M$
$f(D)$	The result of any query operation f
X	$X \sim Lap(\Delta f/\epsilon)$ is a random noise and follows the Laplace distribution of scale parameter $\Delta f/\epsilon$
$S(q)$	The sensitivity of q
$q(D, r)$	q represents a given practical evaluation function. D is data set, r represents any legitimate output
$G(V, E, \chi)$	G is a social network. V represents the set of users in G. $(V_N^U \cup V_N^K = V_N) \cup (V_P^U \cup V_P^K = V_P) = V$. The friendship link denoted by E. The set of user attributes is denoted by χ. $\chi_R \cup \chi_I = \chi$.
T	The set of attribute categories in G, $T_R \cup T_I = T$
L	The attribute value, equal to label, $L_m \subset T_m, l_n \in L_m$ $(1 \le n \le \lvert L_m \rvert)$
$SimA$	The similarity of attributes between users
$\vartheta_{m\in M}^{T_R}$	The accuracy of dependence
ζ	The parameters for evaluating utility, it is close to 1 that means utility better, otherwise, it is close to 0, utility lower

Output

Task 1: Prediction method can predict L^U for negative users who do not know the label of recommended attribute.

Task 2: Publish a noise-added recommendation list.

For the convenience of readers, the symbols involved in the paper are summarized in Table 1.

5. CDAI Method

The framework of this paper is mainly divided into three parts: the first part is the data processing, the second part is the recommendation, and the third part is the privacy protection. In the first part of data preprocessing, the machine learning algorithm is mainly used to find the dependent attribute of the target attribute and delete the weak dependent attribute, so as to reduce the dimension of high-dimensional original data. In the second part, the recommendation combines the classification algorithm of community detection and machine learning to improve the accuracy of prediction. The third part is privacy protection, the differential privacy based on recommended. Differential privacy is constructed for the naive Bayesian algorithm of community discovery and machine learning classification, respectively. After differential privacy treatment, the recommendation results are published to the public, which makes it impossible for the attacker to infer more information about the user in reverse.

5.1. Data Processing. This section uses machine learning to get the dependent attributes of the target attribute. On the one hand, dimensional reduction processing is carried out for high-dimensional data, so as to improve the overall running speed. On the other hand, after deleting the original data, publishing can reduce the information leakage of users from the source and protect their personal privacy.

Input: $G = (V, E, \chi)$;
Output: D_1
(1) **for** $i \in T$ **do**
(2) $T_{del,i}$ use M $\longrightarrow \vartheta_{m\in M}^{T_R}(G)$
(3) $\Delta_{m\in M}^{T_R} = \vartheta_{m\in M}^{T_R}(G)|_T - \vartheta_{m\in M}^{T_R}(G')|_{T'=T-t_i}$
(4) **if** $\Delta_{m\in M}^{T_R} = \max \Delta_{m\in M}^{T_R}$ **then**
(5) $T_i, T_R \longrightarrow high - dependence$
(6) others are low-dependence $\longrightarrow T_{del}$
(7) **end if**
(8) $T' = T - T_{del}$
(9) $T' useM \longrightarrow T_B^U$
(10) **end for**
(11) Get D_1

Algorithm 1: Data processing.

This section uses the machine learning classifier (such as KNN, NB, SVM) to predict the labels of the target attribute. Remove one weak dependency attribute at a time according to formula (9). And use formula (10) to determine whether the deletion will continue. When the utility is reduced by more than ζ, the deletion stops. The remaining attributes are highly dependent. The recommendation accuracy obtained based on this step will be reduced. But by setting the utility, it is reduced to an acceptable range. See Algorithm 1.

5.2. CDAI Recommendation Method. In social networks, users join specific communities according to their preferences. Users in the community may not have any prior

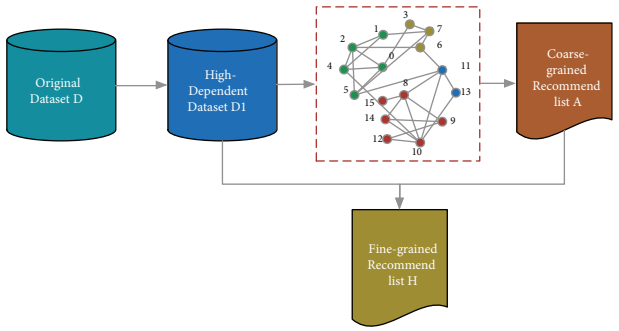


Figure 2: CDAI recommendation structure.

connection to each other. However, these users may have high attribute similarity. From the perspective of sociology, users with high similarity are more likely to have common preferences. Based on this idea, a coarse-grained prediction method based on community detection is proposed. At the same time, even in the same community, there is a high similarity of attributes, but the difference of one of its attribute values still has a specific influence on the final prediction results. For example, in a movie community, both people like comedy movies. However, because of gender differences, a more detailed classification might yield different results. Therefore, in the second stage of the recommendation method, the prediction based on the attribute and machine learning is proposed, to modify the partial prediction results of the first stage. The recommended structure is shown in Figure 2.

This paper is based on the user's friend links and attributes. The community detection algorithm and machine learning classification algorithm are combined to obtain high recommendation accuracy. The recommendation process is divided into two stages. The first stage is a coarse-grained recommendation based on community detection. The Louvain algorithm combined with attribute similarity divides users into communities based on target attribute. The following formula is used to obtain the most probability of target attribute labels in the entire network (see formula (12)), which is assigned to users who are belonging to independent communities and do not have the target attribute label. At the same time, each community has a label that gets the most votes, assigning the label to an empty target attribute in the community. Here, the community weight is added to the probability of getting the labels in each community. The modularity of each community is regarded as the weight of the community (see formula (11)).

$$P_{l_j} = \frac{|l_j|}{|c_i|} Q_{c_i} \tag{11}$$

$$l_{\max j} = \arg\max_{l_j} \sum_{c_i} P_{l_j} \tag{12}$$

In the second stage, the machine learning classification algorithm is used to adjust the prediction results. Thus, higher prediction accuracy can be obtained. The main step of this part is to use the machine learning classification algorithm to obtain the prediction results. In the first step, the predicted results of the independent community remained unchanged. If the predicted results in the community are different from those in the first step, the predicted results of the same community in the first step are replaced by those obtained in the second step. End of recommendation. See Algorithm 2.

5.3. CDAI Privacy Protection and Publishing. It is known from the data preprocessing section that the original data released by social network providers is preprocessed data, while other data is hidden data, which is not visible to the public. Based on this assumption, the maximum range of background knowledge obtained by an attacker is the data published by a social network provider, that is, preprocessed data. For this part of the dataset, this paper uses differential privacy to protect the dataset when the attacker knows all about it. The structure of this part is consistent with that of the recommendation part and consists of two parts. The structure is shown in Figure 3. The first part is a differential privacy based on community detection. The second part is the differential privacy based on Naive Bayes [52].

5.3.1. Differential Privacy Based on Community Detection. This part of differential privacy is based on community discovery. To achieve the effect of privacy protection, it is necessary to hide the link relationship between users in the community detection. Therefore, Laplace noise is added to the community weight to hide the link relationship in the community. Add noise to the number of links to node I within and outside the community to hide the true link relationship. Privacy budget is ϵ. Function sensitivity Δf is related to function. The number of users within the community is the maximum sensitivity of K_{in}. The number of users outside the community is the maximum sensitivity of $\sum_{tot}$. Therefore, the combined sensitivity of the two sequences is the total number of users in the social network. Add noise to formula (2) and change it to the following form:

$$Q^*_{c_i} = \frac{\sum_{in} + lapace\left(k_{i,in}/\epsilon\right)}{2b} - \left(\frac{\sum_{tot} + lapace\left(k_i/\epsilon\right)}{2b}\right)^2 \tag{13}$$

Algorithm 3 achieves the ϵ-differential privacy protection by adding Laplace noise. Therefore, it is necessary to prove that the algorithm strictly abides by the ϵ-difference privacy definition. Each step of the algorithm will be analyzed and proved according to the definition and properties of Section 3.

Differential Privacy Proof. The algorithm adds Laplace noise to each link weight of nodes in Q_{c_i}. We know from the formula for Q that Q is made up of two parts. The first part is c_i internal weight. When only one node u_r is added, Node u_r is connected to $|c_i|$ users in c_i. Node u_r is connected to $d_i n$ users in c_i, and d is at most equal to all $|c_i|$ users in c_i. The second part is c_i external weight. The external maximum number of users is $|V|-|c_i|$. Because it adds Laplace to each link weight of the nodes, when two data sets differ by a tuple, the sensitivity in c_i is $k_{i,in}$. If the maximum weight is 1, the sensitivity is $|V|$. Similarly, the maximum sensitivity outside c_i is also $|V|$. Therefore, ΔQ_{u_r} satisfies $\epsilon/|V|$ -differential privacy. We know from the formula for Q that each user u_r joins is independent of the other. So it is parallel relationship. According to the

Input: $G = (V, E, \chi)$;
$V = Users$;
$E = Links$;
$\chi = Attribute\ set$;
L^K =The known labels of users($L^K = \{l_i \in T_R \cup T_I \mid u_i \in V^K, V^K = V^N \cup V^P\}$);
Output: L^U =The unknown labels of users($L^U = \{l_i \in T_R \mid u_i \in V^U, V^U = V - V^K\}$);
(1) **Phase 1:** Get the coarse-grained $L^U_{coarse-grained}$, List A
(2) Get a new social network graph G'
(3) $SimA(i, j) = |x_i \cap x_j|/|X|$
(4) $G' = G \bigcap SimA$
(5) Initiallize Each Node to a Separated Community
(6) **repeat**
(7) **for** $i \in |V|$ **do**
(8) **for** $j \in |V|$ **do**
(9) Remove i from its community, place to j's community
(10) Compute the composite modularity gain Δ
(11) **end for**
(12) Choose j with maximun positive gain (if exists) and move i to j's community
(13) Otherwise i stays in its community
(14) **end for**
(15) **until** No further improvement in modularity
(16) Gets the label probability in each community, $P_{l_j} = (l_j/c_i)Q_{c_i}$
(17) Find $P_{max} \longrightarrow l_{max}$, Assign l_{max} to $L^{U_{in}}(U_{in} \in c_i)$
(18) $l_{\max j} = \arg\max_{l_j} \sum_{c_i} P_{l_j}$
(19) Assign $l_{\max j}$ to $L^{U_{independent}}(U_{independent} \in c_i)$
(20) Get the coarse-grained L^U_A
(21) **Phase 2:** Get the fine-grained $L^U_{fine-grained}$ for the unknow labels of users in community. Get the list B.
(22) **for** $j \in |V|$ **do**
(23) **if** $A_j \neq B_j$ **then**
(24) $A_j = B_j$
(25) Users of the independent community, the labels remain the same
(26) **end if**
(27) **end for**
(28) Get final L^U

Algorithm 2: CDAI recommendation method.

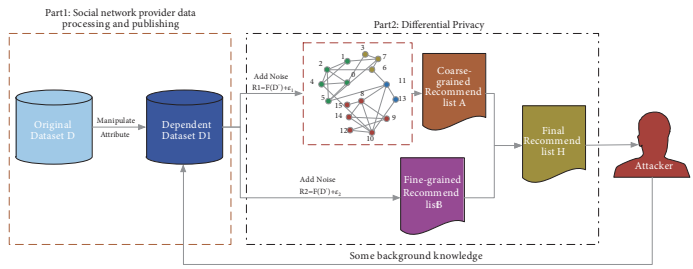


Figure 3: CDAI privacy protection structure.

parallel property of differential privacy, Q_{c_i} satisfies $\epsilon/|V|$ - differential privacy.

5.3.2. Differential Privacy Based on Naive Bayes. Based on the second part of the recommendation, the machine learning classification algorithm is used to predict labels. In this paper, KNN, NB, and SVM are mainly used. This part only constructs differential privacy for naive Bayesian algorithm. Generally, an NB classifier has a good effect on data classification, so it can be considered that it has a good estimate of joint probability density $P(X, Y)$. So, for a randomly generated feature vector x, if the classifier divides it into some class of c_j, we can think of (x, c_j) as a random sample from $P(X, Y)$. The process of selecting the maximum posterior probability class c_{max} is Naive Bayes. It is regarded as the process of voting according to conditional probability and prior probability. According to the exponential mechanism, a decision algorithm in accordance with differential privacy protection is designed. The output of the algorithm is a class variable. To reduce the effect of parameters on the posterior

Input: c_i = Divided communities in G,$c_i \in C$;
L^K = Known labels of users($L^K = \{l_i \in T_R \cup T_I \mid u_i \in V^K, V^K = V^N \cup V^P\}$);
Output: L^{*U} = Unknown noise labels of users($L^U = \{l_i \in T_R \mid u_i \in V^U, V^U = V - V^K\}$);
$L^{*U_{in}=}$ Unknown noise labels of users, users in community i. $U_{in} \in c_i$
$L^{*U_{independent}=}$ Unknown noise labels of users, users in independent community i. $U_{independent} \in c_i$
(1) **for do**$c_i \in |C|$
(2) **for do**$l_j \in |L^U|$
(3) $P^*_{l_j} = (|l_j|/|c_i|)Q^*_{c_i}$
(4) **end for**
(5) **end for**
(6) Find the max probability $P^*_{\max l_j}$ in Community i, $\max l_j \longrightarrow L^{*U_{in}}$
(7) $l_{\max j} = \arg\max_{l_j} \sum_{c_i} P_{l_j} \longrightarrow L^{*U_{independent}}$
(8) Get L^{*U}_A

Algorithm 3: Differential privacy based on community detection.

probability, Laplace smoothing of conditional probability is carried out, and the following definitions of utility function are given:

$$P_\lambda\left(X^{(j)} = a_{jl} \mid Y = c_k\right) = \frac{\sum_{i=1}^{N} I\left(x_i^j = a_{jl}, y_i = c_k\right) + \lambda}{\sum_{i=1}^{N} I\left(y_i = c_k\right) + S_j\lambda} \tag{14}$$

N is the number of samples, $I(x)$ is the indicator function. $x^{(j)_j}$ is the jth feature vector. S_j is the number of values of $x^{(j)_j}$. When $\lambda = 1$, it is called Laplace smoothing.

$$q\left(D, x, c_k\right) = \log\left(p\left(c_k\right)\right) + \sum_{i=0}^{n} \log\left(p\left(x^{(i)} \mid c_k\right)\right) \tag{15}$$

Theorem 15. *For any two random samples x and x' that differ by only one attribute, $q(D, x, C_k)$ has a local sensitivity of* $\log(2N)$. *N is the number of tuples in dataset D.*

Proof. In a given dataset, the conditional probability of $p(x_i^{(j)} \mid c_k)$ after Laplace smoothing is bounded. Its upper bound is 1. $\sum_{i=1}^{N} I(x_i^{(j)} = a_{jl}, y_i = c_k) \geq 0, \sum_{i=1}^{N} I\ (y_i = c_k) \leq N, S_j \leq N, \lambda = 1$. When the inequality equals, the conditional probability is minimized $p(x_i^{(j)} \mid c_k) = 1/2N$.

$$p\left(x_i^j \mid c_k\right) \in \left[\frac{1}{2N}, 1\right) \tag{16}$$

□

Given the data set D, for the n dimensional feature vectors x and x' with only the jth dimension different, the only thing that is not equal is the conditional probability of the jth dimension. Namely, $q(D, x, c_k) - q(D, x', c_k) = \log(p(x^{(j)} \mid c_k)) - \log(p(x'^{(j)} \mid c_k))$. According to the formula above,

$$\left|q\left(D, x, c_k\right) - q\left(D, x', c_k\right)\right| < \log\left(2N\right) \tag{17}$$

By Theorem 15, the sensitivity of the utility function Δq only related to the size of the data set N. Replacing global sensitivity with local sensitivity will leak information related to the size of the dataset. Here $\Delta q = \lceil \log(2N) \rceil$. Because the logarithm function is a slow growing function, therefore, the uplift operation can reduce the possibility of data set size information leakage. Finally, the label c_{max} is selected with probability $p_{c_r} \propto \exp(\epsilon q(D, x, c_r)/2\Delta q)$. Satisfy ϵ-difference privacy.

According to the serial nature of differential privacy. The difference privacy based on community discovery is concatenated with the difference privacy based on Bayesian. It satisfies $(|V| + 1)\epsilon/|V|$ -differential privacy.

6. Evaluation

6.1. Datasets. We will compare Attribute-Link and CDAI using real dataset. In our experiment, we investigate two different datasets. The first one is Facebook dataset (https://snap.stanford.edu/data/egonets-Facebook.html) and the second one is Google+ dataset (https://snap.stanford.edu/data/egonets-Gplus.html). Both of them have the ego-network and profile of each user and anonymized the information. From Facebook we got profile and network data from 10 ego-networks and consist of 88,234 links and 4,039 users with an average circle size of 22 friends. Each user has own profile, which has 22 attribute categories. Profile includes birthday, education classes, education school, education year, and hometown. From Google+ we obtained profile and network data from 132 ego-networks. It consists of 13,673,453 links and 107,614 users. 132 ego-networks represented 132 users. The Google+ dataset is quite different to those from Facebook, in the sense that their creators have chosen to release them publicly. It contains 6 attribute categories which, respectively, are gender, institution, job, last name, city, and university. In experiment, we choose the single value attribute categories as inherent attributes and the multiple-value attribute categories regard as recommendation attributes. Table 2 provides the general statistics of the two datasets. This table shows that all of the two graphs are almost fully connected.

6.2. Experiment Settings. We describe the metrics adopted to evaluate various inference, training and testing, and parameter settings.

Table 2: General statistics about the two datasets.

Network Property	Facebook	Google+
Number of nodes	4039	107614
Number of friendship links	88234	13673453
Number of attribute categories	22	6
Number of recommend attribute categories	13	4
Number of inherent attribute categories	9	2
Diameter (longest shortest path)	8	6
Average clustering coefficient	0.6055	0.4901

Evaluation Metrics. The aim of this paper is to provide recommendation service for users. At the same time, it prevents attackers from using background knowledge and recommendation content to push back other information of users. Since we set the target user is negative user. Due to the evaluation, assume all test users are negative users. Randomizing a multiple-value attribute category is recommended attribute category and predicts the attribute values, for example, we know that Facebook has 22 attribute categories from Table 2. There are 13 recommended attribute categories and 9 inherent attribute categories; therefore, randomly choose one from recommended attribute categories as the target attribute to predict. Use the rest of attribute categories to find the relationship with chosen recommend attribute. We use these models to predict: (1) Attribute-Link inferring attributes model; (2) CDAI model.

In this section, given 3 evaluation metrics, respectively, accuracy, modularity, and utility, the community detection assesses using modularity. The general range of modularity is between -0.5 and -1. When the range is between 0.3 and 0.7, the clustering effect is good. The rest of metrics evaluates the prediction result. Accuracy is the ratio of the number of samples correctly to the total sample size for a given test data set. The high-dependence in Section 4 is based on accuracy. The formula is as follows.

$$Accuracy = \frac{l_{correct}}{L_{total}} \tag{18}$$

where $l_{correct}$ is denoted by the number of correct labels obtained using the classifiers. L_{total} is the number of test samples.

In addition, the utility evaluation metric is discussed in Section 4.5; we will use the definite formulas for evaluation; see formula (10). In this paper, we set ζ as 0.2. The classifiers include KNN, Naive Bayes, and SVM. The ϵ is 0.001, 0.01, 0.05, 0.1, 0.5, 1,1.5, 10, 100, 1000, 10000.

Train Dataset and Test Dataset. Facebook dataset consists of 10 ego-networks, and Google+ dataset includes 133 ego-networks. In this experiment randomly select the train dataset and the test dataset, and the ratio is 8 to 2. We assume all users are target users in test dataset. Each user's attributes consist of recommended attributes and inherent attributes. The target attribute is randomly selected from recommendation attribute categories. Due to the profile obtained in real world, therefore, some information is missing. These links between centre user and his followers are nature weak relationship in network graph, extracting high-dependence links from each ego-network and dividing the community based on the attribute similarity.

Table 3: The modularity of two datasets.

	Link-Att	CDAI
Facebook	0.52896	0.52428
Google+	0.19456	0.30578

6.3. No Noise Labels Predicted. In this section, we compared Attribute-Link method and CDAI method, exploiting modularity, accuracy, and utility to evaluate the two methods. Table 3 shows the comparison results of initial modularity between the two methods.

From Table 3, no matter which datasets the modularity of CDAI is relatively stable and the results are better than Attribute-Link method. Figures 4(a)–4(c) are the test results of Google+ dataset; no matter which classifiers the prediction result of CDAI is significantly better than the other one. In the classifier of KNN in Figure 4(a), the accuracy does not change when using inherent attributes or recommended attributes. In the process of removing the attributes, when removing the sencon attribute category, classification method based on Attribute-Link improved its accuracy; meanwhile using CDAI method the result has no change that means the latter one has better stability than the former. From Figure 4(b) of NB classifier, when only using inherent attributes for prediction, the accuracy improved. However, when deleting the third attribute category, the accuracy of two methods all improved. The SVM classifier has always been smooth in the figure. Figures 4(d)–4(f) are the results of Facebook dataset. There is a larger decline when only using single attribute categories (inherent attribute categories) that means the recommended attribute categories have a great impact for prediction. In NB classifier, no matter which attribute is deleted, the accuracy is stability. But the other classifiers have a sharp decline when the attribute deleted the fourteenth. The results of CDAI in three pictures are always better than Attribute-Link.

Figures 5(a)–5(f) are the utility results of Google + and Facebook datasets. Based on the definition of utility, the values are closer to 0, and the utility is better. From Figure 5, either data or classifier, the two methods on the utility results are below setting values; at the same time the results obtained from CDAI are better than the Attribute-Link. Figures 5(a)–5(c) are the results of Google +. Using KNN classifier, when removing the second attribute category based on Attribute-Link method, the utility is suddenly increased that illustrates that there is bigger influence on the utility. However, based on CDAI, no matter which result is removed, the attribute categories have stabilization. In Figure 5(b) using NB classifier, the two methods have same line trend and when removing the third attribute category there has a big change. For Figure 5(c) based on SVM classifier, the results are smooth and steady use Attribute-Link method and CDAI method.

The results of Facebook dataset are shown in Figures 5(d)–5(f). In the three subfigures, when deleting

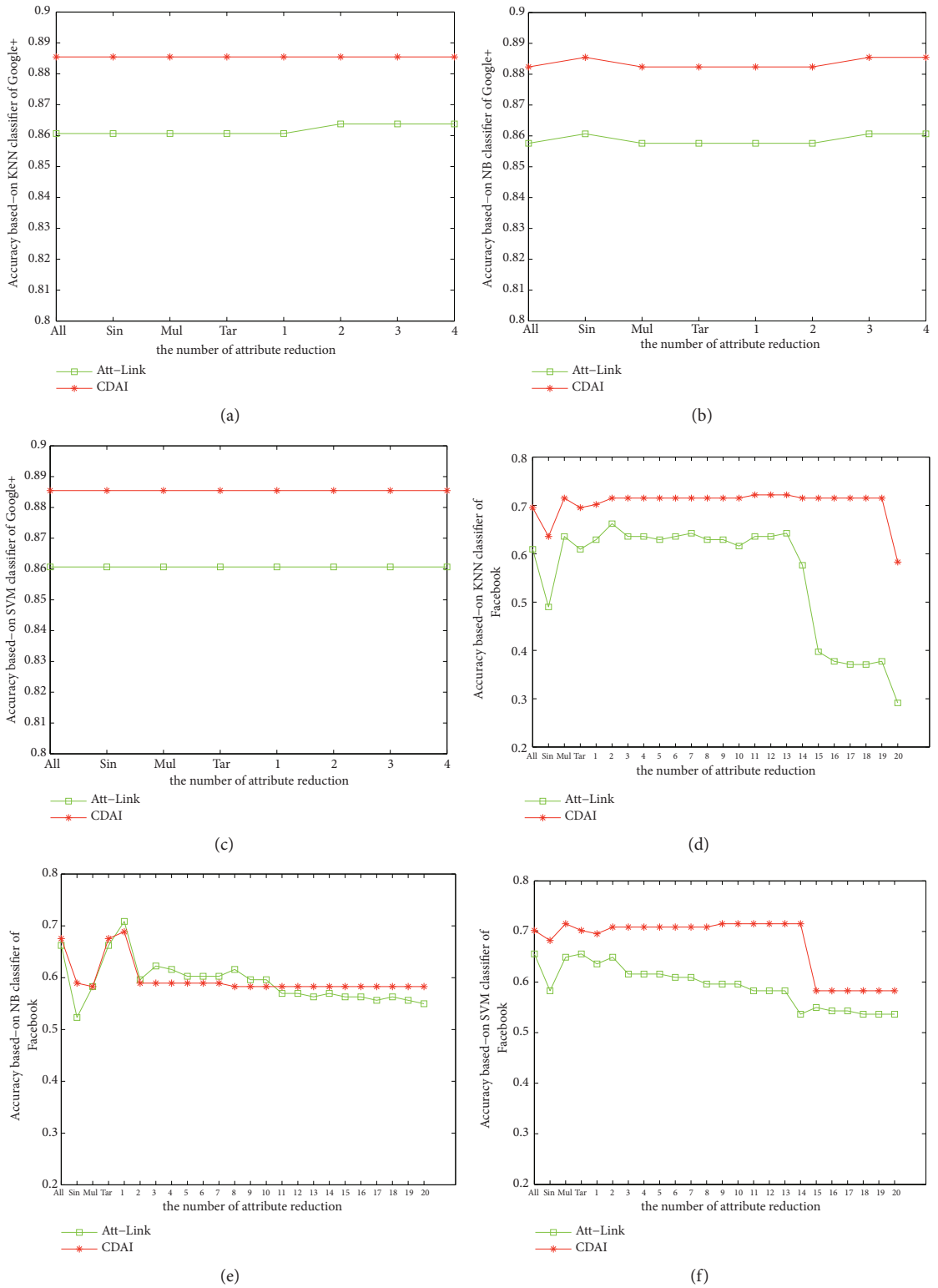


FIGURE 4: The accuracy of predict based on Google+ dataset and Facebook dataset. Figures (a)-(c) are the results of Google+ dataset. Figures (d)-(f) are the results of Facebook dataset.

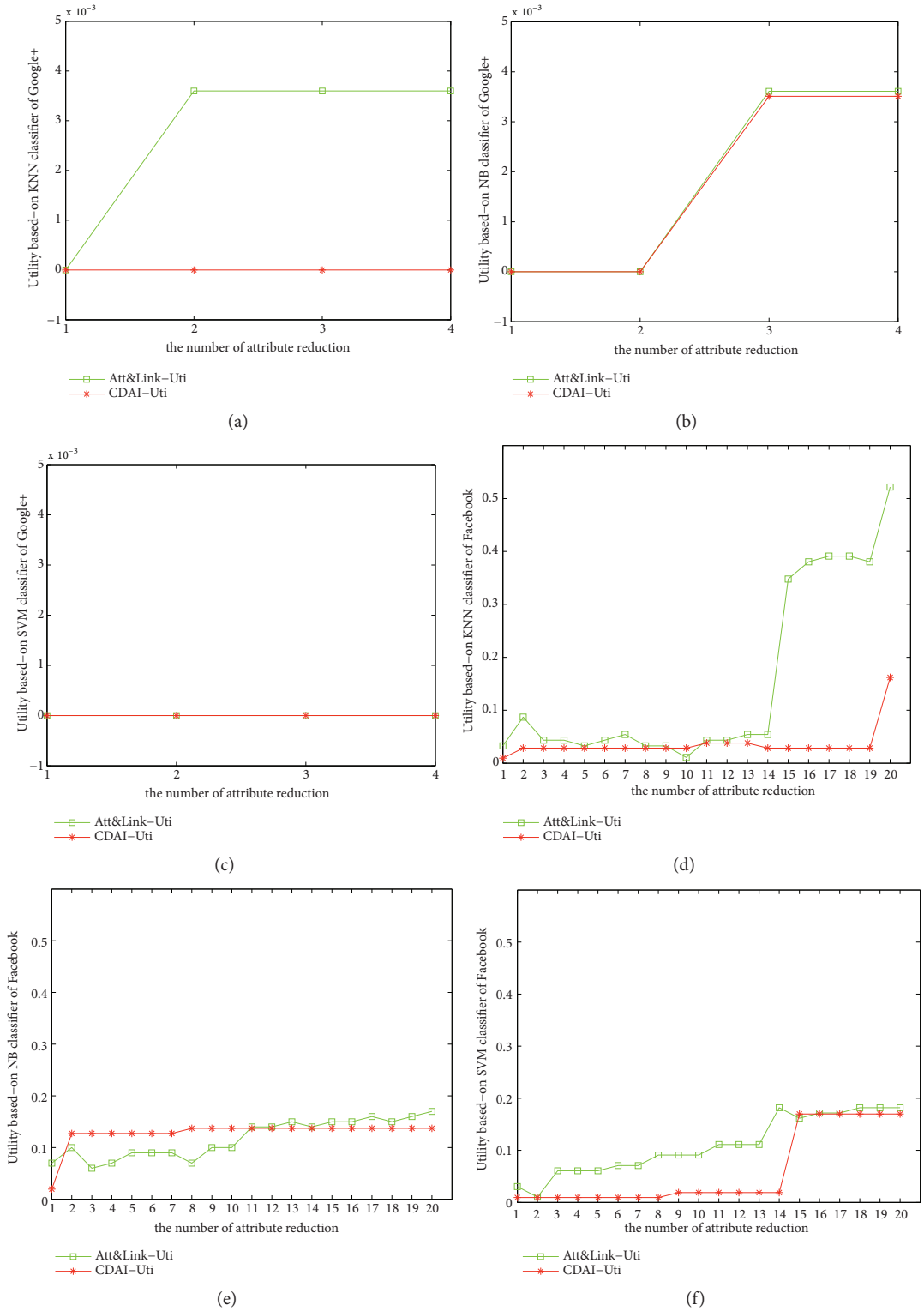


FIGURE 5: The utility based on Google+ dataset and Facebook dataset. Figures (a)-(c) are the results of Google+ dataset. Figures (d)-(f) are the results of Facebook dataset.

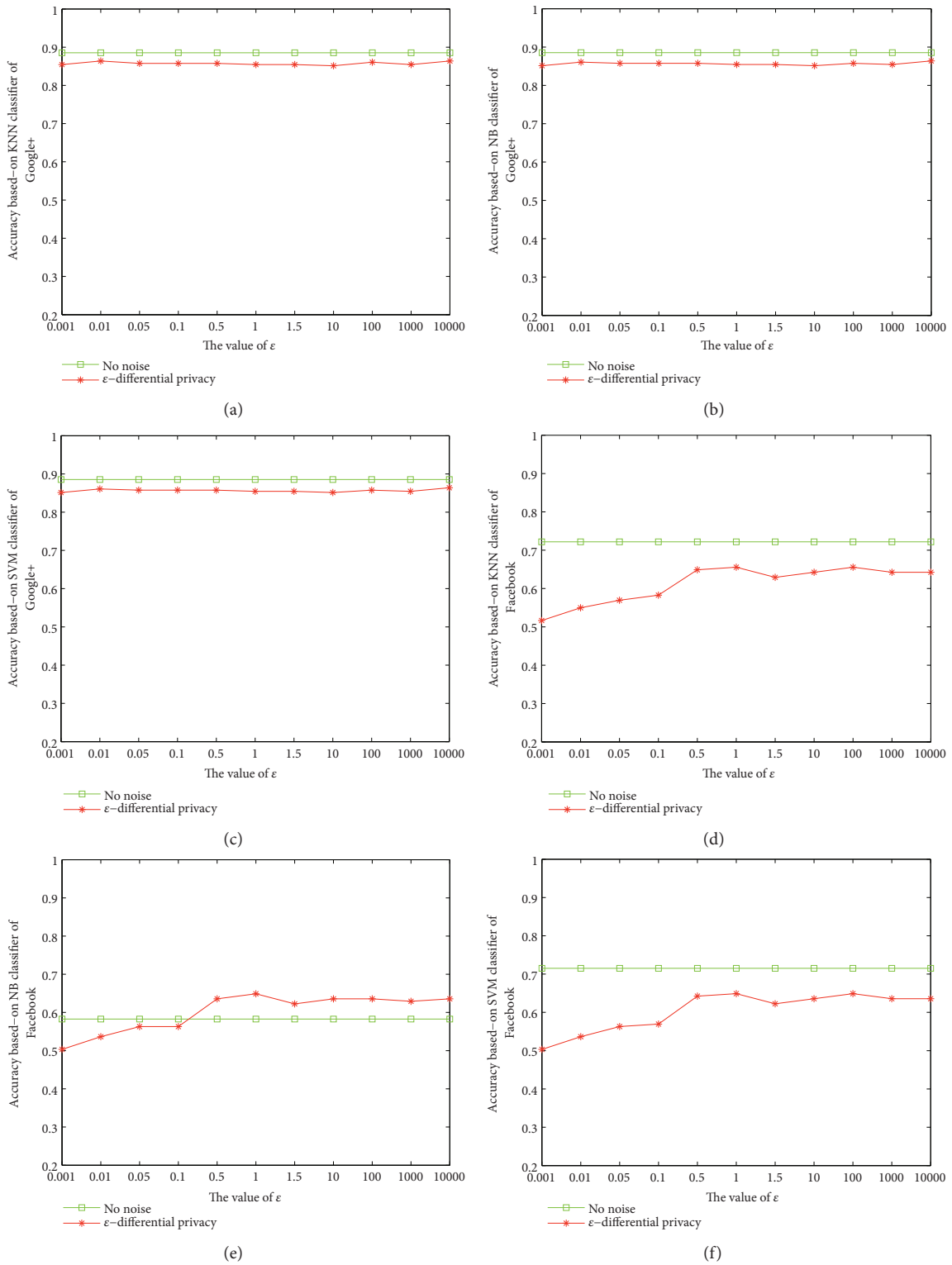


FIGURE 6: The accuracy with noise based on Google+ dataset and Facebook dataset. Figures (a)-(c) are the results of Google+ dataset. Figures (d)-(f) are the results of Facebook dataset.

the second attribute category the utility changed. But based on the definition and set value of utility, this change did not influence the removal. When deleting the fifteenth attribute category, for Figures 5(d) and 5(f) which show huge change, the change range is outside the set value; therefore stop removing the rest of attribute categories.

6.4. Comparison with Noise. Figures 6(a)–6(f) show the predicted results after adding noise. Figures 7(a)–7(f) show

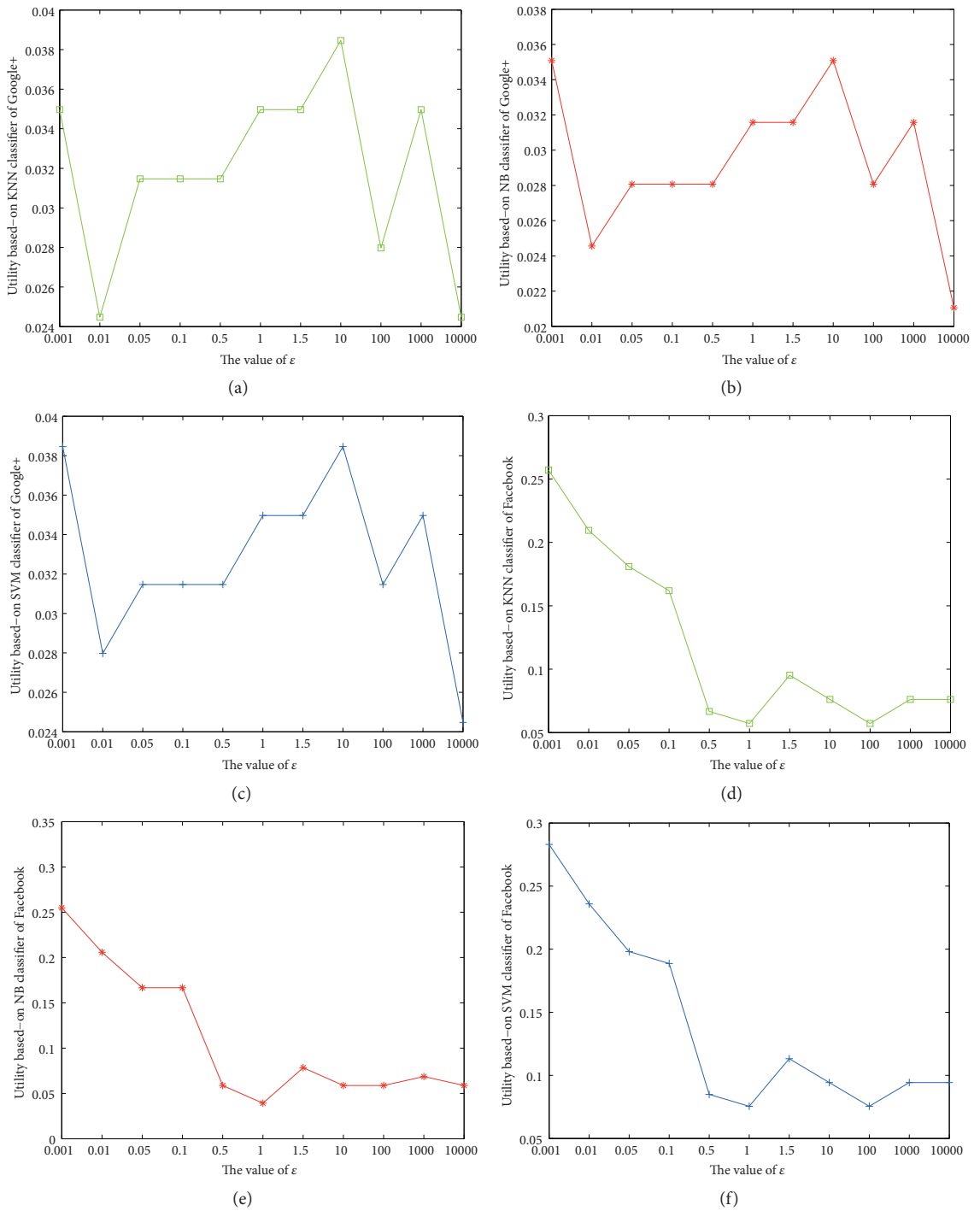


FIGURE 7: The utility with noise based on Google+ dataset and Facebook dataset. Figures (a)-(c) are the results of Google+ dataset. Figures (d)-(f) are the results of Facebook dataset.

the utility after adding noise. The first three graphs in Figures 6 and 7 are based on Google+ dataset. It can be seen from the figures that the noise has certain influence on the prediction and the utility is reduced to a certain extent. However, due to the characteristics of the dataset itself, the effect is not significant. Because every attribute in Google dataset has a very large attribute value, the attribute value of a single attribute remains large even after many attributes have been deleted. Therefore, even if the maximum noise is added, the results are relatively stable.

The last three graphs in Figures 6 and 7 are based on Facebook dataset. As can be seen from the figure, noise has a greater impact on prediction, although the reduced utility is within a given range. However, it can be seen that the prediction results are in accordance with differential privacy. When ϵ increases, the accuracy of the prediction

also increases, and the result is close to that without adding noise.

The reason why Facebook's results are more obvious than Google's is that Facebook's attribute types are diverse and each attribute type has less value, so the effect of adding noise is obvious. However, in general, it still conforms to the protection of differential privacy, and the reduced utility is within the range, reaching the balance between privacy protection and utility.

7. Conclusions

This paper proposes a CDAI recommendation method based on community detection and user attributes. The experiment is based on real social network data. It is found that this method can effectively improve the accuracy of recommendation compared with traditional classification methods. At the same time, in order to prevent the published recommendation content from being used by the attacker to push back users' privacy information, this paper also proposes a novel privacy protection method. First, this method manipulates on nodes and links. Combining with differential privacy and publishing the final recommendation results, this method will lose a few of utility. But it can protect users' privacy. This paper also has some limitations when facing attacks from attackers with complete background knowledge. This is the author's future work.

Data Availability

The data used to support the findings is generally unavailable due to public releasability constraints. However, please contact the corresponding author for special release consideration.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partly supported by the NSF under grant No. 1252292, No. 1829674, No. 1741277, No. 1704287 and the National Natural Science Foundation of China under Grant No. 61632010, No. 61502116, No. U1509216, No. 61370217, No. 61472096, No. 61872105.

References

[1] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems, RecSys 2016*, pp. 191–198, September 2016.

[2] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, "A hybrid collaborative filtering model with deep structure for recommender systems," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 1309–1315, February 2017.

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[4] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[5] S. M. McNee, J. Riedl, and J. A. Konstan, "Being accurate is not enough: how accuracy metrics have hurt recommender systems," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI EA '06)*, pp. 1097–1101, Montreal, Canada, April 2006.

[6] S. Vargas and P. Castells, "Rank and relevance in novelty and diversity metrics for recommender systems," in *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys '11)*, pp. 109–116, Chicago, Ill, USA, October 2011.

[7] K. Lewis, M. Gonzalez, and J. Kaufman, "Social selection and peer influence in an online social network," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 109, no. 1, pp. 68–72, 2012.

[8] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *The Computer Journal*, vol. 42, no. 8, pp. 30–37, 2009.

[9] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS '07)*, pp. 252–260, Vancouver, Canada, December 2007.

[10] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*, pp. 203–210, ACM, Boston, MA, USA, July 2009.

[11] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM'08*, pp. 931–940, USA, October 2008.

[12] B. Yang, Y. Lei, J. Liu, and W. Li, "Social Collaborative Filtering by Trust," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017.

[13] "Spear phishing attacks".

[14] P. Gupta, S. Gottipati, J. Jiang, and D. Gao, "Your love is public now: Questioning the use of personal information in authentication," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, ASIA CCS 2013*, pp. 49–59, May 2013.

[15] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "BlurMe: Inferring and obfuscating user gender based on ratings," in *Proceedings of the 6th ACM Conference on Recommender Systems, RecSys 2012*, pp. 195–202, September 2012.

[16] E. Zheleva and L. Getoor, "To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles," in *Proceedings of the 18th International World Wide Web Conference, WWW 2009*, pp. 531–540, April 2009.

[17] A. Chaabane, G. Acs, M. A. Kaafar et al., "You are what you like! information leakage through users' interests," in *Proceedings of the 19th Annual Network & Distributed System Security Symposium (NDSS)*, 2012.

[18] M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 110, no. 15, pp. 5802–5805, 2013.

[19] N. Z. Gong, A. Talwalkar, L. MacKey et al., "Joint link prediction and attribute inference using a social-attribute network," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 2, article 27, 2014.

[20] Neil. Zhenqiang Gong and Bin. Liu, "You are who you know and how you behave: Attribute inference attacks via users social friends and behaviors," in *Proceedings of the USENIX Security Symposium*, pp. 979–995, 2016.

[21] J. Jia, B. Wang, L. Zhang, and N. Z. Gong, "Attriinfer: Inferring user attributes in online social networks using markov random fields," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 1561–1569, Perth, Australia, April 2017.

[22] N. Z. Gong and B. Liu, "Attribute inference attacks in online social networks," *ACM Transactions on Privacy and Security*, vol. 21, no. 1, 2018.

[23] Y. Lin, X. Wang, F. Hao, L. Wang, L. Zhang, and R. Zhao, "An on-demand coverage based self-deployment algorithm for big data perception in mobile sensing networks," *Future Generation Computer Systems*, vol. 82, pp. 220–234, 2018.

[24] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1-1, 2017.

[25] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *Proceedings of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, May 2017.

[26] J. Lu, Z. Cai, X. Wang, L. Zhang, P. Li, and Z. He, "User social activity-based routing for cognitive radio networks," *Personal and Ubiquitous Computing*, pp. 1–17, 2018.

[27] 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. 99, pp. 1-1, 2017.

[28] M. Humbert, T. Studer, M. Grossglauser, and J.-P. Hubaux, "Nowhere to hide: Navigating around privacy in online social networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8134, pp. 682–699, 2013.

[29] D. Jurgens, T. Finethy, J. McCorriston, Y. T. Xu, and D. Ruths, "Geolocation prediction in twitter using social networks: A critical analysis and review of current practice," in *Proceedings of the 9th International Conference on Web and Social Media, ICWSM 2015*, pp. 188–197, May 2015.

[30] A. McCallum, K. Nigam et al., "A comparison of event models for naive bayes text classification," in *Proceedings of the AAAI-98 workshop on learning for text categorization*, vol. 752, pp. 41–48, Citeseer.

[31] X. Zheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Follow but no track: privacy preserved profile publishing in cyber-physical social systems," *IEEE Internet of Things Journal*, 2017.

[32] B. Carminati, E. Ferrari, and A. Perego, "Security and privacy in social networks," in *Social Computing: Concepts, Methodologies, Tools, and Applications*, pp. 1706–1717, IGI Global, 2010.

[33] L. Sweeney, "*k*-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[34] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "Resisting structural re-identification in anonymized social networks," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 102–114, 2008.

[35] N. Vuokko and E. Terzi, "Reconstructing randomized social networks," in *Proceedings of the 10th SIAM International Conference on Data Mining, SDM 2010*, pp. 49–59, May 2010.

[36] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4–7, 2006. Proceedings*, vol. 3876 of *Lecture Notes in Computer Science*, pp. 265–284, Springer, Berlin, Germany, 2006.

[37] F. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," in *Proceedings of the International Conference on Management of Data and 28th Symposium on Principles of Database Systems, SIGMOD-PODS'09*, pp. 19–30, July 2009.

[38] R. Chen, N. Mohammed, B. C. M. Fung, B. C. Desai, and L. Xiong, "Publishing setvalued data via differential privacy," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 1087–1098, 2011.

[39] Y. Liang, Z. Cai, Q. Han, and Y. Li, "Location privacy leakage through sensory data," *Security and Communication Networks*, vol. 2017, Article ID 7576307, 12 pages, 2017.

[40] J. Wang, Z. Cai, Y. Li, D. Yang, J. Li, and H. Gao, "Protecting query privacy with differentially private k-anonymity in location-based services," *Personal and Ubiquitous Computing*, pp. 1–17, 2018.

[41] X. Zheng, G. Luo, and Z. Cai, "A Fair Mechanism for Private Data Publication in Online Social Networks," *IEEE Transactions on Network Science and Engineering*, 2018.

[42] 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.

[43] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep Learning Based Inference of Private Information Using Embedded Sensors in Smart Devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[44] 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*, pp. 1-1, 2018.

[45] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, "Cost-efficient strategies for restraining rumor spreading in mobile social networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2789–2800, 2017.

[46] Z. He, Z. Cai, and X. Wang, "Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks," in *Proceedings of the 35th IEEE International Conference on Distributed Computing Systems (ICDCS '15)*, pp. 205–214, July 2015.

[47] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Article ID P10008, 2008.

[48] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 2, Article ID 026113, 2004.

[49] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proceedings of the 48th Annual Symposium on Foundations of Computer Science (FOCS '07)*, pp. 94–103, Providence, RI, USA, October 2007.

[50] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 6, Article ID 066133, 2004.

[51] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, pp. 24-24, Atlanta, Ga, USA, April 2006.

[52] X. Chen, J. Liu, X. Feng, and X. Zhao, "Differential privacy synthetic data set publishing algorithm based on naive bayes," *Computer Science*, vol. 42, pp. 236–238, 2015.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 7608198, 12 pages
https://doi.org/10.1155/2018/7608198

Research Article

Towards a Novel Trust-Based Multicast Routing for VANETs

Hui Xia,[1,2] **San-shun Zhang**,[1] **Ben-xia Li**,[1] **Li Li**,[1] **and Xiang-guo Cheng**[1]

[1]*College of Computer Science and Technology, Qingdao University, Qingdao 266071, China*
[2]*Department of Computer Science and Technology, Shandong University, Qingdao 266237, China*

Correspondence should be addressed to Hui Xia; xiahui@qdu.edu.cn

Received 12 June 2018; Revised 20 August 2018; Accepted 5 September 2018; Published 1 October 2018

Guest Editor: Chunqiang Hu

The Intelligent Transportation System (ITS) is an important application area of the Cyber-Physical System (CPS). To further promote effective communication between vehicles, vehicular ad hoc networks (VANETs) have been widely used in the ITS. However, the communication efficiency in VANETs is not only affected by the external environment but also more vulnerable to malicious attacks. In order to address the above-mentioned issues, we propose a novel trust-based multicast routing protocol (TMR) to defend against multiple attacks and improve the routing efficiency. In the proposed trust model, direct trust is calculated based on Bayesian theory and indirect trust is computed according to evaluation credibility and activity. The fuzzy logic theory is used to fuzzify the direct and indirect trust values, and then the total trust value of the node is obtained by defuzzification. With the help of the obtained trust values, malicious vehicle nodes are eliminated in the processes of route establishment and route maintenance, and finally, the network establishes trusted and efficient routes for data delivery. Comprehensive simulation experiments show that our new protocol can effectively improve the transmission rate of data packets at the expense of a slight increase in end-to-end delay and control overhead.

1. Introduction

As a combination of computation processes and physical processes, cyber-physical systems (CPS) organically integrate cyber systems and physical systems from environmental perception [1]. To monitor road conditions, reduce traffic accidents and improve transportation safety, researchers have applied CPS to the smart transportation system to form the Transportation Cyber-Physical System (T-CPS) [2]. With the popularity of wireless communication technology and mobile computing, the vehicular ad hoc network (VANET) is assumed to be a significant component for both safety and infotainment applications in T-CPS. VANET mainly adopts a vehicle-to-vehicle (V2V) communication mode, where participating vehicles and other neighboring vehicles are allowed to exchange data by the wireless transceiver, and if necessary, packets can be routed via adjacent vehicles to destinations outside the communication range. Drivers can make correct decisions by exchanging information about each other's driving intentions while driving. Emergency vehicles can warn cars to take prompt evasive actions, thereby providing an emergency passage. Vehicles can adjust routes in time according to congestion alerts issued by the traffic monitoring system. In the research on VANET, data distribution is regarded as one of the most critical processes, which mainly depends on efficient routing protocols [3]. In order to meet the different requirements of traffic users, the routing protocols for VANET should have specific constraints, such as a lowering of the end-to-end delay and packets loss when forwarding packets. So far, researchers have proposed routing protocols to ensure reliable data transmission and information exchange between vehicles, which are roughly divided into two categories (geography-based and topology-based) according to different networking methods [4].

Compared with the static or low-speed moving nodes in the traditional wireless network, the VANET node moves faster and unpredictably, which leads to frequent changes of network topology. Besides, owing to the unique features of VANET itself, such as its dynamism, complexity, and uncertainty, the difficulty of routing protocol design in VANET is increased still further. On the other hand, the VANET routing protocol is more vulnerable to threats due

to the lack of infrastructure and the self-organization of the network. Malicious vehicles may incorrectly forward and even drop packets, or divert packets towards the wrong relay nodes, preventing data from reaching their destination nodes. Furthermore, vehicles colluding in the network may also falsely improve their reputation, or maliciously slander trusted vehicles, thereby interfering with the assessment of node trustworthiness. If the reliability of the data cannot be adequately assessed, the driver may make wrong judgments based on malicious information received from misbehaving vehicles, resulting in severe traffic jams. To solve the routing security problem, cryptography-based and trust-based security mechanisms have been proposed [5–8]. The former focuses on the integrity of the data but does not identify malicious entities within the network [7]. On the contrary, the trust-based security mechanisms are widely used in dealing with internal attacks from malicious entities [8, 9]. However, how to design a secure and efficient routing protocol on the basis of the trust model is still a tremendous challenge in VANET.

Therefore, in this paper, we first design a novel trust model based on fuzzy logic theory to compute the trust value of vehicles by direct trust and indirect trust. Then, a secure multicast routing protocol based on the trust model is presented to deal with multiple attacks (e.g., black attack, grey attack, slander attack) from malicious vehicular nodes. The main contributions of this paper are as follows:

(1) In the calculation of trust values of nodes, various influencing factors affecting the trust are fully considered. Also, the time decay function is used to ensure that the node's recent performance is given higher weight in the trust calculation.

(2) The fuzzy logic theory is used to fuzzify the value of the real decision factor, fuzzy inference is performed according to fuzzy inference rules, and the final trust value of the node is obtained after defuzzification.

(3) We integrate trust models into routing protocols and design a secure multicast routing protocol to achieve reliable and secure communication from multiple sources to multiple destinations. Furthermore, we give detailed descriptions of the data structure and packet format, route discovery process and route maintenance process.

The remainder of this paper is organized as follows. Section 2 discusses recent works in the literature. We describe in detail a trust model in Section 3. In Section 4, we propose a trust-based multicast routing protocol. The experimental results are shown in Section 5. Finally, Section 6 presents concluding remarks and directions for future research.

2. Related Works

A routing protocol is the premise of effective operation for VANET. Slavik et al. [10] presented the Distribution-Adaptive Distance with Channel Quality (DADCQ) protocol which could select forwarding nodes based on the distance method. The decision threshold function is created to determine the decision threshold value and avoid the influence of node density, spatial distribution pattern, and wireless channel quality. Li et al. [11] proposed a novel intersection-based routing with QoS support for VANET, which composed of terminal intersection selection process, network exploration process and optimal routing path selection process. To minimize the end-to-end delay, Togou et al. [12] developed a distributed routing protocol called Stable CDS-Based Routing Protocol (SCRP). Before sending data messages, the protocol computes end-to-end delay of entire routing path based on the weight of each road segment. A cooperative volunteer protocol was proposed for VANET in [13] to broadcasting warning messages to emergency vehicles in Non-Line of Sight (NLOS) conditions. Moreover, the context-aware system is established to collect data and detect NLOS situations. To respond quickly to emergencies and prevent further accidents, Mezher et al. [14] designed a multimedia multi-metric map-aware routing protocol. The real maps with SUMO are utilized to create a realistic scenario, and the REVsim tool is exploited to choose a right forwarding node. In [15], the authors improve the existing ant colony optimization protocol using the concept of fuzzy logic, in which the validity of link is calculated by bandwidth, received signal strength metric and congestion metric.

With the increasingly prominent problem of route security, several schemes based on trust management have been proposed to deal with malicious attacks in VANET. Marmol et al. designed a proposal based on trust and reputation infrastructure for VANET in [16], which can precisely identify malicious or selfish nodes to prevent the spread of false or bogus messages. In [17], an event-based reputation system (EBRS) was proposed for defending against the conspired Sybil attacks. EBRS can detect and suppress the propagation of false messages employing reputation and trusted value of events. The probabilistic and deterministic approaches are used together to assess trust in [18]. The former computes the trust level of vehicles by using collected information, while the latter measures the trust level based on distances calculated. Sun et al. [19] presented a trust evaluation model based on membership cloud model for VANET. The goal of cloud model precisely describes the uncertainty of the trust relationships. Besides, they compute the cloud droplets and the aggregated trust values using trustworthiness and algorithm. Dahmane et al. [20] introduced a weighted probabilistic and trust-aware strategy to ensure that the most reliability relay nodes are selected in multi-hop communications. The authors in [21] proposed a security framework that consists of a hybrid trust model and a misbehaviour detection system, to detect malicious vehicles.

In recent years, some researchers have integrated the trust model into routing protocols and proposed many trust-based secure routing protocols. Based on the intersection-based routing protocol GyTAR, Bouali et al. developed a novel secure routing protocol S-GyTAR in [22]. The protocol monitors vehicles by cluster-based mechanism and evaluates trust value by a reputation-based scheme. A trusted routing protocol based on GeoDTN+Nav is proposed in [23], primarily, Bayesian trust management model and opportunistic routing forwarding models are used to establish security paths. Kerrache et al. [24] demonstrated a trust-based routing protocol, named TROUVE, to find the shortest and safest routing to a destination by the distribution of dishonest nodes in VANET.

Gazdar et al. [25] established distributed trust computing framework for VANETs, which employed the direct experience to calculate neighbors' trust value and utilized a tier-based approach to alleviate malicious behaviours. Given the authority levels of nodes, Yao et al. [26] developed a weight-based dynamic entity centre trust model. Furthermore, they also integrate the model into the routing protocol GPSR to enhance the security and improve the data delivery rate.

However, these above-mentioned researchers have not comprehensively investigated how to manage trust in vehicle ad hoc networks in a holistic manner. Besides, those mechanisms can introduce excessive routing overhead. Moreover, so far as we know, there are fewer security protocols for multicast which consider the concept of trust.

3. Trust Model

In this section, we describe our trust model in detail. The trust model in this paper contains two parts: the calculation of node trust and the calculation of path trust. The node's trust is determined by two trust factors: direct trust and indirect trust. And the path's trust is determined by the trust values of all the nodes on the path. Moreover, in this paper, we introduce a novel method to synthesize a relevant node's trust value.

3.1. Calculation of Node Trust. We introduce the sliding window mechanism to make sure that the latest interaction period can take a greater weight in the calculation of node trust.

3.1.1. Direct Trust. This latest period T can be divided into t time periods, that is, $T = \{T_1, T_2, T_3, \ldots, T_t\}$. It is assumed that in each period T_i, node A can monitor n_i times the forwarding behaviors of node B. The packet-forwarding ratios of node B in time period T_i are $v_{n_i} = \{v_{n_1}, v_{n_2}, v_{n_3}, \ldots, v_{n_i}\}$. We set a threshold ($L$) for the packet-forwarding ratio, if there are m_i times higher than the threshold L, and then the number below L is n_i-m_i. We make use of the Bayes theorem to reduce the error in computing the reputation of a specific node, which is calculated by combining the previous reputation level and likelihood function (i.e., the posterior probability in Bayes theorem is equal to the prior probability multiplied by the adjustment factor). Beta distribution conforming to the binomial distribution has the property of the conjugate prior. Applying the beta distribution to the Bayes theorem, if the prior probability satisfies the Beta distribution and the binomial distribution function denotes the likelihood function, then the final posterior probability is also satisfied the Beta distribution. In this way, we can keep the form of prior probability and posterior probability constant, and give both the prior probability and the posterior probability clearly physical meanings, separately.

In this paper, the last reputation vector denotes the prior probability and the current reputation vector represents the posterior probability. At this moment, the binomial distribution function can be used as the adjustment factor. The general form of the Beta distribution function is $\beta(a, b) = x^{a-1}(1 - x)^{b-1}/B(a, b)$, where $B(a, b)$ is a normalization constant which is used to ensure that the total probability is 1. We set $B(a, b) = \int_0^1 x^{a-1}(1 - x)^{b-1}dx$, where x satisfies $[0, 1]$. We assume that $R_{i\text{-}1}$ represents the value of the $(i\text{-}1)^{\text{th}}$ reputation evaluation vector according to the Beta distribution, and then the calculation of the posterior probability is $R_i = (f_i(x)/\int_0^1 f_i(x)R_{i-1}dx)R_{i-1}$. In this paper, two mutually independent events that conform to binomial distribution are taken as whether the forwarding rate is higher than the threshold value L, then it can obtained $f_i(x) = C_{n_i}^{m_i} x^{m_i}(1 - x)^{n_i - m_i}$. The reputation evaluation vector sequence of node A to B is $R_n = \{R_1, R_2, R_3, \ldots, R_t\}$. As mentioned above, if $R_{i\text{-}1} \sim beta(a_{i\text{-}1}, b_{i\text{-}1})$, then $R_i \sim beta(a_{i\text{-}1} + m_i, b_{i\text{-}1} + n_i\text{-}m_i)$. Then the values $a_1 = m_1$, $b_1 = n_1\text{-}m_1, \ldots, a_i = a_{i\text{-}1} + m_i$, $b_i = b_{i\text{-}1} + n_i\text{-}m_i$. Therefore, at the beginning t_0, we can set $a_0 = b_0 = 1$, $R_0 \sim beta(1, 1)$. One of the trust metrics t_d in the T_i time period can be computed and updated by using the expectation of beta distribution. The calculation equation is shown as follows:

$$t_d = \frac{a_i}{a_i + b_i} \quad (i \geq 1) \tag{1}$$

The direct trust of node B from the point of view of node A should also be affected by their interaction time and the total amount of packets forwarded. Therefore, we can use two metrics to describe the above-mentioned conditions, i.e., the time attenuation factor and the total amount of packets forwarded factor. The equation for calculating the direct trust is shown as follows:

$$dt_{AB} = \frac{\sum_{i=1}^{t} \rho^{t-i} M_{AB}^{i}}{M} * t_d \tag{2}$$

where $M = \sum_{i=1}^{t} \rho^{t-i}$, ρ^{t-i} $(0 < \rho < 1)$ denotes the time attenuation function, and M_{AB}^{i} denotes the amount of packets forwarded by node B for node A during the T_i period. The greater the amount of packets forwarded is, the higher the confidence of the obtained trust value will be. When there is no direct interaction between nodes A and B, the value of dt_{AB} is set to 0.5.

3.1.2. Indirect Trust. The indirect trust factor is crucial in the calculation of node trust. If node A and node B have no historical interactions, node A can still calculate the trust value of node B based on gathering other nodes' trust recommendations of node B. In this paper, two kinds of trust recommendation metrics (i.e., recommendation credibility and activity factor) are used to calculate the indirect trust of a specific node.

(1) Recommendation Credibility. Some malicious nodes do not discard or modify packets, while maliciously defaming other trustworthy nodes. This type of attack is also called a bad-mouth attack or slander attack. In order to resist these attacks, we can calculate the recommendation credibility of the recommended nodes. It is assumed that $\{j_1, j_2, j_3, \ldots, j_r\}$ are the neighbors of node j which have interactions with node i [27]. We can also divide the interaction period into $N * T_t$ $(N >= 1)$ time periods during which node i interacts with

any recommender. The direct trust of node j_l within the time period T_k is $dt^k_{ij_l}$ ($1 <= k <= N$) evaluated by node i, and the direct trust of j_l is $dt^k_{jj_l}$ evaluated by node j. Then, after the N-th interaction, the degree of deviation of node j for the trust evaluation of node j_l is calculated using the following equation:

$$tr_{ijj_l} = \sum_{k=1}^{N} \rho^{N-k} \left(1 - \left|dt^k{}_{jj_l} - dt^k{}_{ij_l}\right|\right) \tag{3}$$

Supposing the number of common nodes is r, these nodes interact with both node i and node j. Then compared with node i, the total degree of deviation of node j for trust evaluation is as follows:

$$tr_{ij} = \frac{\sum_{l=1}^{r} tr_{ijj_l}}{r} \tag{4}$$

We can suppose that there are R nodes that interact with j's neighbors. The greater the value of R is, the more accurate the obtained trust value will be. Then the recommendation credibility of node j is as follows:

$$tr_j = \frac{e^{-1/R} \sum_{i=1}^{R} tr_{ij}}{R} \tag{5}$$

where $0 < e^{-1/R} < 1$, and this metric is used to adjust the number of nodes on the evaluation of trust. The greater the value of R is, the closer it is to 1. We can also obtain a higher recommendation credibility value for node j. Moreover, we set an acceptable threshold for the recommendation credibility REC_THRESHOLD. If the node's recommendation credibility is lower than this threshold, the recommended information supplied can be ignored.

(2) Activity Factor. The metric H can be defined as the activity factor. If the number of neighbors of a specific node is G, and the number of neighbors that have recently interacted with this node is F, then we can calculate roughly the activity factor of this node using the following equation, $H = F/G$.

Finally, we can obtain the indirect trust of a specific node B as follows:

$$re_B = \frac{\sum_{j=1}^{F} tr_j dt_{jB} M_{jB}}{\sum_{j=1}^{F} tr_j M_{jB}} * H_B \tag{6}$$

Sections 3.1.1 and 3.1.2, respectively, calculate the node's direct trust and indirect trust. It is difficult to find an accurate mathematical model to integrate these two factors. Yet the ability to close the gap between imprecise human reasoning and the computational logic of fuzzy logic makes it especially attractive for the trust evaluation of the nodes.

3.1.3. Synthesis of Node Trust. The fuzzy logic model used in this paper is similar to the traditional fuzzy logic system which contains the following four steps: (1) transform the true value variable into a fuzzy set by a fuzzifier; (2) design fuzzy IF-THEN rules; (3) ue the fuzzy inference engine combined with fuzzy IF-THEN rules to derive the node's degree of trustworthiness; (4) use a defuzzifier to convert the fuzzy trustworthiness output into the real trust value.

First of all, we divided the two fuzzy trust factors into two levels, that is, great and small.

Definition 1 (input or output range). $0 \le dt \le 1$, $0 \le re \le 1$. The closer the value is to 1, the greater the input is. The output degree of node's trust is divided into four levels: fully trustworthy (ft), overall trustworthy (ot), generally trustworthy (gt), untrustworthy (ut).

Definition 2 (node trust value). The trust value of a node $0 \le$ trust ≤ 1. The closer the value is to 1, the higher the trustworthiness of the node is. Then these member functions are defined using triangular membership functions as follows:

$$small(x) = \begin{cases} \dfrac{0.6 - x}{0.6} & x \in (0, 0.6) \\ 0 & x \in (0.6, 1) \end{cases} \tag{7}$$

$$great(x) = \begin{cases} 0 & x \in (0, 0.4) \\ \dfrac{x - 0.4}{0.6} & x \in (0.4, 1) \end{cases} \tag{8}$$

$$ft(t) = \begin{cases} \dfrac{t - 0.6}{0.4} & t \in (0.6, 1) \\ 0 & t \in (0, 0.6) \end{cases} \tag{9}$$

$$ot(t) = \begin{cases} \dfrac{t - 0.4}{0.2} & t \in (0.4, 0.6) \\ \dfrac{0.8 - t}{0.2} & t \in (0.6, 0.8) \\ 0 & \text{others} \end{cases} \tag{10}$$

$$lt(t) = \begin{cases} \dfrac{t - 0.2}{0.2} & t \in (0.2, 0.4) \\ \dfrac{0.6 - t}{0.2} & t \in (0.4, 0.6) \\ 0 & \text{others} \end{cases} \tag{11}$$

$$ut(t) = \begin{cases} \dfrac{0.4 - t}{0.4} & t \in (0, 0.4) \\ 0 & t \in (0.4, 1) \end{cases} \tag{12}$$

The x in (7) or (8) represents the dt or re value of a specific node. The t in (9)~(12) denotes the trust value of this node. Input fuzzy set membership functions and output fuzzy set membership functions can be shown in Figure 1.

Because of the subjective characteristic of trust, the node is more inclined to believe in the empirical value based on the direct interactions between it and other nodes. Therefore the fuzzy IF-THEN rules we designed should ensure that the dt value takes a higher weight in the calculation of node's trust values. According to this, we set the four fuzzy IF-THEN rules as follows:

(1) IF dt is great AND re is great, THEN the node is fully trustworthy.

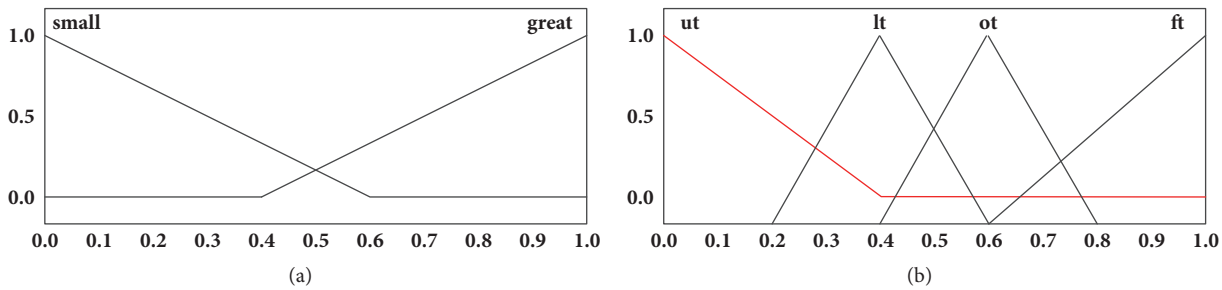


Figure 1: Membership functions: (a) input fuzzy set membership functions; (b) output fuzzy set membership functions.

(2) IF dt is great AND re is small, THEN the node is overall trustworthy.

(3) IF dt is small AND re is great, THEN the node is generally trustworthy.

(4) IF dt is small AND re is small, THEN the node is untrustworthy.

The membership functions and rules given above are just a set of instances. They can be revised as needed. If the input and output sets can be more reasonable, the evaluation result could be more objective.

The output fuzzy set is calculated by the fuzzy inference engine, and we define some required operators before building the engine. Then we apply the centroid calculation method to defuzzify the output fuzzy node trust value and finally get the true trust value of a node. The centroid calculation equation is shown as follows:

$$trust_B = \frac{\int_0^1 tf\,(t)\,dt}{\int_0^1 f\,(t)\,dt} \tag{13}$$

where $f(t)$ is the final expression of the trust value through fuzzy reasoning. We test several sets of trust values and get the results in Table 1.

We put four kinds of combination, i.e., $dt = 0.5$, $re = 0.4$; $dt = 0.4$, $re = 0.5$; $dt = 0.6$, $re = 0.4$; $dt = 0.5$, $re = 0.8$, into the fuzzy logic system to get the trust values of the node in Figure 2. We can figure out that the value of dt has a greater influence on the calculation of node's trust value than re through analysing a large number of results.

Figure 3 shows the distribution of trust values of a specific node for different input combination of dt and re. From the trend in the curvature of this graph, apparently we can see that when the dt value of the node is more than 0.4, as the dt value increases, the trust value of the node increases rapidly.

3.2. Calculation of Path Trust. The credibility of a routing path should also be assessed. In this paper, we use the metric 'path trust' (PT) to evaluate the above content. The path trust value is closely related to the trust value of each intermediate node on the path [28]. We refer to the barrel theory and take the node trust value which has the lowest credibility as the path trust value. For example, suppose that there are n nodes on the path, the source node is S, and the destination node is D. Then the path trust value PT_{SD} is calculated as follows: $PT_{SD} = \min\{trust_S, \ldots, trust_D\}$.

4. Applying Trust Enhancement to Multicast Routing

In this section, we apply trust enhancement to the standard multicast ad hoc on-demand distance vector routing protocol (i.e., MAODV [29]). We call this new trust-based routing protocol as multicast trust-based ad hoc on-demand distance vector routing protocol (i.e., MTAODV). Any node in the network can calculate its neighbor's trust value, and it can select a trustworthy routing path for data delivery.

4.1. Trust-Based Route Discovery. Three new fields are added to the original RREQ messages (i.e., Table 2) of MAODV that are reverse path trust, required path trust, and malicious node address. The initial value of reverse path trust is 1. If a node wants to join the multicast group but has no valid route, it will broadcast a J_flag RREQ message. The reverse path is built when the RREQ message comes to the reply node. A node that is close to the required node denotes the upstream node. In contrast, it is a downstream node if it is close to the reply node. The node that receives the message can calculate the trust value of the sending or forwarding message node [30, 31]. This relevant node trust value will be used to compare with the path trust value, and the reverse path value will be updated to the smaller one. However, if the node trust value is smaller than the required path trust, the J_flag RREQ message will not be forwarded further.

One new field (i.e., average trust value, ATV) is added to the original RREP messages (i.e., Table 3) of MAODV. Assume that a selected routing path contains n nodes, and then the average trust value can be calculated using the following equation:

$$ATV = \frac{\sum_{i=1}^{n} trust_i}{n}, \tag{14}$$

where $trust_i$ is the trust value of any node on the path. The multicast group member who has received the J_flag RREQ message will reply with the RREP to the source node. The forwarding route is built when the source node receives the message. When there is more than one path from the source node to the destination node, the source node should activate one of them. The traditional MAODV protocol stipulates that the shortest one is selected as a priority [32]. In MTAODV, the trust factor is the most important. So the destination node will choose a path that has the greatest average trust value

Table 1: The inference results of node's trust values.

dt	*re*	*Trust*
0.9	0.3	0.6
0.3	0.9	0.4
0.6	0.4	0.464
0.5	0.8	0.562

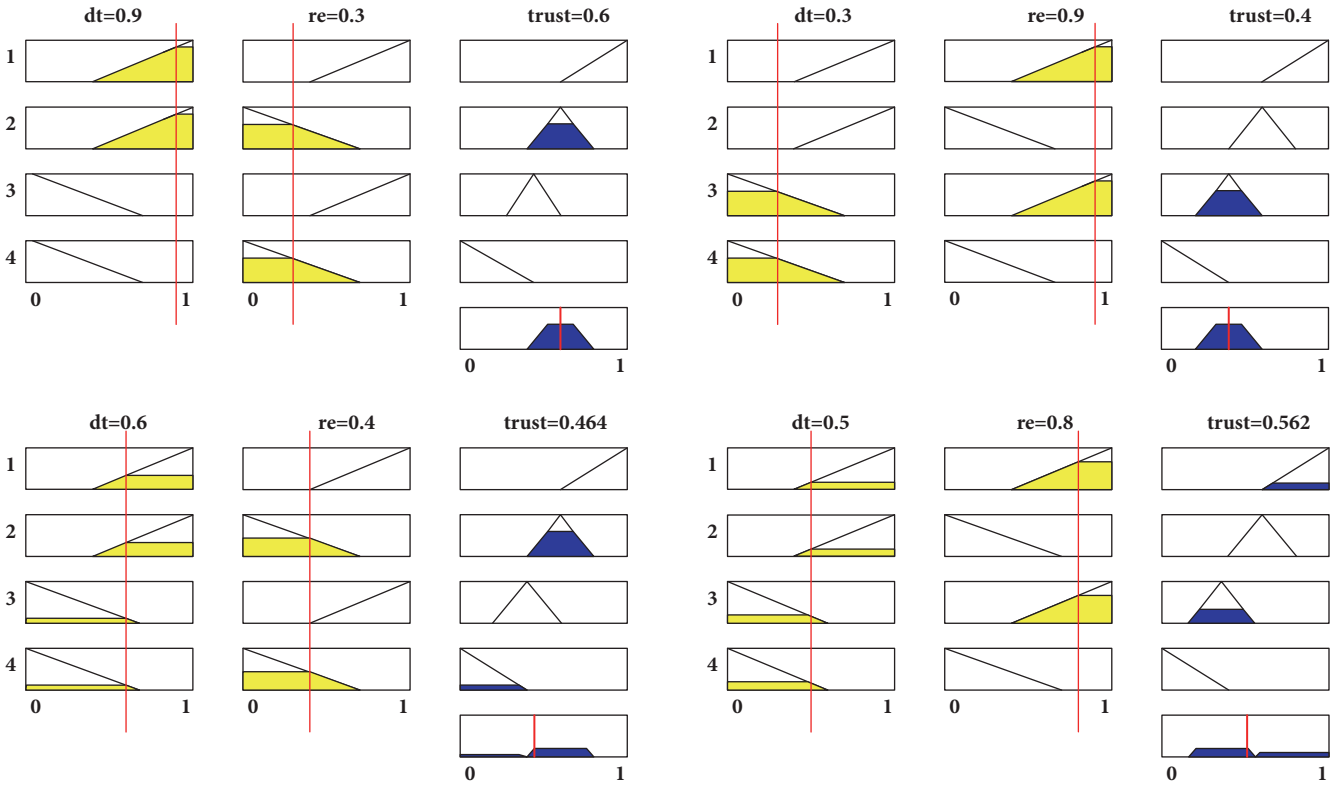


Figure 2: Node trust evaluation process.

Table 2: Enhanced RREQ message.

Dest_Addr
Dest_Seq
J_flag
R_flag
Originator IP Address
Originator Sequence Number
Lifetime
Reverse Path Trust
Required Path Trust
Malicious Node Address

Table 3: Enhanced RREP message.

Originator IP address
Dest_Addr
Dest_Seq#
R_flag
Mgroup_hop
Lifetime
Hop_Cnt
Average Trust Value

to send a MACT message. The path that has received the message is activated, and any node that has not received the message will delete the path of its cache.

4.2. Trust-Based Route Maintenance. Each multicast group member maintains a multicast routing table. In this paper, we put all the malicious node addresses in an array and place the array in a multicast routing table, as shown in the Table 4 'enhanced multicast routing table'.

After the multicast group is set up and the data is being transmitted, the upstream node can monitor the forward behaviors of the downstream node. If the downstream node is detected as a malicious node, the upstream node will unicast an RREQ message with this malicious node address (as shown in the Table 5) to the group leader. The group leader receives the message and replies with an RREP message to

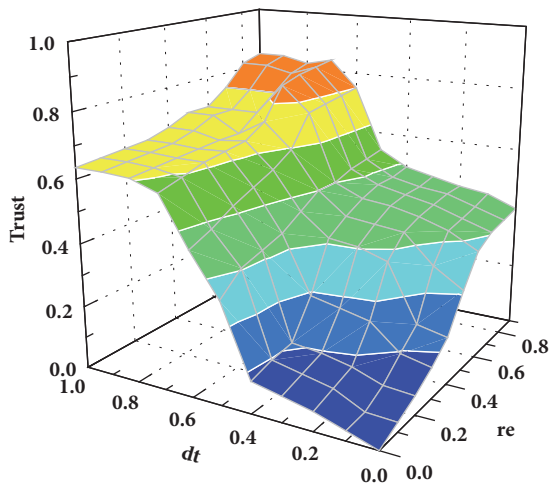


Figure 3: The surface of the node trust value.

Table 4: Enhanced multicast routing table.

Multicast Group IP Address
Multicast Group Leader IP Address
Multicast Group Sequence Number
Hop Count to Multicast Group Leader
Hop Count to next Multicast Group Member
Next Hops: Next Hop IP Address; Next Hop Interface; Link Direction; Activated Flag
{*MaliciousNode*$_1$ *Address*; *Malicious Node*$_2$ *Address*; *Malicious Node*$_3$ *Address*; }

that node. Then the group leader broadcasts a group hello message with the malicious node address to the entire network. A node that receives the message will record the malicious address in its multicast routing table. All multicast group members will disconnect from this malicious node and rediscover another route to the multicast group. The malicious node cannot be a group member until it recovers from the multicast routing table. It will recover from the multicast routing table after V_Threshold_time, and its trust value will be set to 0.5.

4.3. Restoration of Broken Branches. As shown in Figure 4, node S is the group leader. Supposing node B detects its downstream node C is a malicious node, it unicasts an RREQ message with this malicious node address (i.e., node C) to the group leader (i.e., node S). This group leader will reply with an RREP message to node B after it receives the RREQ message. Then the group leader broadcasts a group hello message with the malicious node address to the entire network. Each node in this network that receives the message will record the malicious address in its multicast routing table. All multicast group members in this multicast tree (i.e., node B and node

Table 5: Enhanced group hello message.

Group Leader IP address
Multicast Group IP address
Multicast Group Sequence Number
U_flag
O_flag
Hop Count
Malicious Node Address

F as shown in Figure 4) disconnect from this malicious node. For instance, node F automatically sends a MACT message with P_flag to the upstream node to disconnect itself from node C. Then, node F rediscovers another route to the multicast group via performing the restoration of broken branches. Besides, since the malicious node information (i.e., node C) has been stored in the malicious node table of the network node, the RREQ message with J_flag (initiated by node F) reforwarded by node C will be ignored by the next hop. This mechanism can ensure that the malicious node can no longer join the tree or become the intermediate forwarding node.

We set {hello_interval $*$ (1 + allowed_hello_loss)} as the threshold time to detect branch disconnection, and use the hello message to detect the break. If a node does not receive a hello message from a neighbor within the time specified above, it can determine that the branch is broken. Once the branch disconnection is detected, the downstream node repairs the broken link. The downstream node broadcasts a RREQ message with J_flag until the message reaches any group member, in which the destination address in this message is set as the address of the group leader, the destination sequence number is set as the last acquired multicast group sequence number, and Mgroup_Hop is set as the number of hops to the group leader. Then the downstream node establishes a path to this specific group member to complete the link repair.

5. Experimental Results

5.1. Experimental Setup. We use the NS-2.35 simulator [33] to estimate the performance of MAODV [29], MTAODV, LWT-MAODV [20] and RBTM-MAODV [26] under different scenarios. The experimental parameters are set as shown in the Table 6 [34]. In a 1000$*$1000 square metre area, 40 nodes perform the random way model. There are two source nodes and eight destination nodes. The delivery ratio threshold L is set to 0.7 and the base of the time attenuation function is set to 0.9. We change the node maximum speed and the number of malicious nodes in the network, respectively. In the following scenarios, two simple types of routing attacks (i.e., grey-hole attacks and black-hole attacks) are launched by malicious nodes. In grey-hole attacks, data packets were selectively forwarded by malicious nodes at a rate of 45%, while in black-hole attacks, all data packets were dropped.

5.2. Performance Evaluation. The performance of the entire network is shown by the following three vectors: packet delivery ratio, end-to-end delay, and control overhead. The

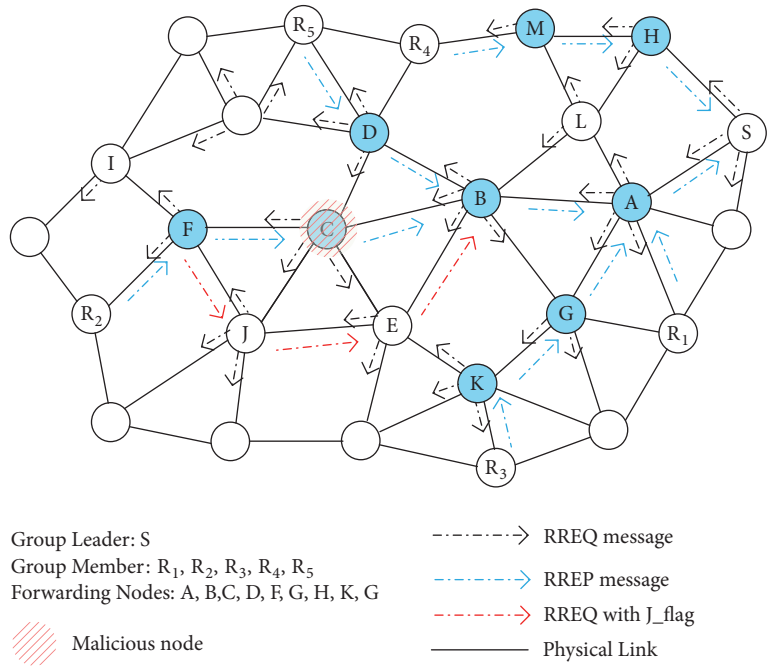


Figure 4: A simple example for restoration of broken branches.

Table 6: Network configuration parameters.

Parameter	Value
Simulation time	3000 s
Topology area	1000 m∗1000 m
MAC protocol	IEEE 802.11
Packet size	256 bytes
Node movement speed	5~30 m/s
Channel bandwidth	2 Mbits/s
Number of nodes	40
Source node	2
Destination node	8
Number of malicious nodes	0~10

packet delivery ratio represents the routing efficiency, and it is the fraction of the data packets that are received by the destination nodes compared to those sent by the source nodes. The end-to-end delay is the time used to receive all packets from the source node to the destination node. The control overhead is the ratio of the control bytes to the packet bytes in a network. Each experiment was repeated 30 times. The simulation results are shown in Figures 5 and 6.

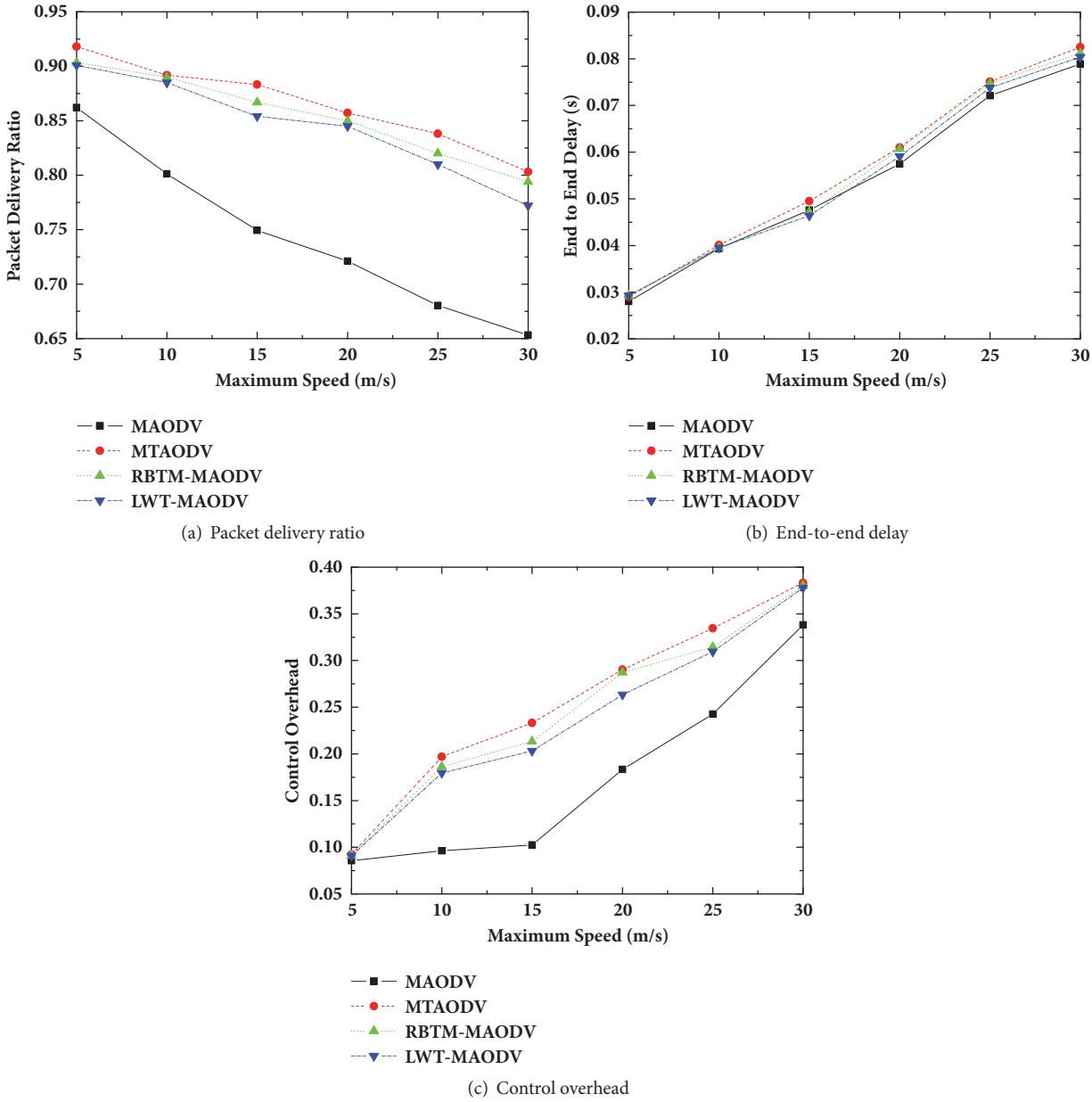
Test 1: Varying Node Maximum Speed. As shown in Figure 5(a), when there are three malicious nodes in the network, the packet delivery ratio is reduced by changing the network node maximum speed. The reason is that, with the nodes moving faster, the network topology changes more frequently, resulting in the probability of the packet transmission path being disconnected increases. The delivery ratio of MAODV declines sharply, while MTAODV, LWT-MAODV, and RBTM-MAODV perform more stable and the MTAODV performs better than the other two trust-based protocols. This figure shows that applying a trust model to a protocol can effectively exclude malicious nodes in the network. Compared with the other two trust models, using more trust metrics to calculate the trust value of nodes makes the MTAODV more effective.

Figure 5(b) shows that the end-to-end delay increases since the route linkage are susceptible to collapse. Along with the increase in node maximum speed, the end-to-end delay in MTAODV, LWT-MAODV and RBTM-MAODV is higher than MAODV. There can be several reasons: (1) These trust-based protocols choose a trusted path instead of the shortest one; (2) Once a malicious node is found on the path, the routing path will be broken. The network will subsequently perform the route maintenance operation, leading to an increase in the end-to-end delay.

Figure 5(c) shows that the control overhead in MTAODV, LWT-MAODV and RBTM-MAODV is relatively high compared with MAODV along with the increase in node maximum speed. The reason is that the enhanced trust-based protocol increases the computational complexity and requires more control packets. All the operations involving trust increase the control overhead.

Test 2: Varying the Number of Malicious Nodes. In the second experiment, we set the node maximum speed to 10 m/s.

Figure 6(a) illustrates that when the number of malicious nodes increases, the packet delivery ratios reduce in the four protocols. The larger the number of malicious nodes, the greater the hop count of the trusted route will be. This results in a notable decrease in the packet delivery ratio. The MAODV decreased significantly while the other



(a) Packet delivery ratio

(b) End-to-end delay

(c) Control overhead

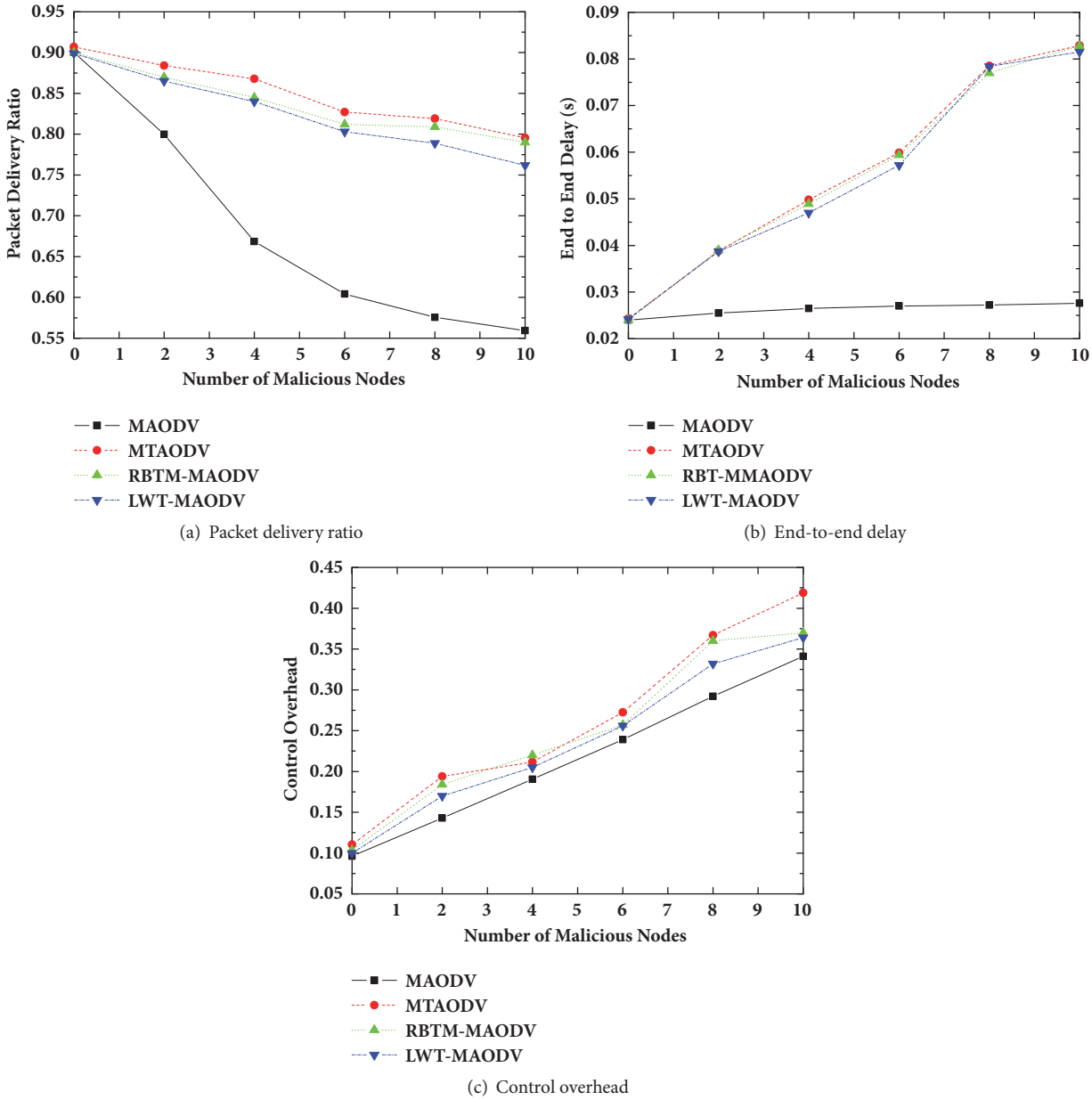
Figure 5: Performance with varying node maximum speeds.

three reduced slowly. The MTAODV performs the best. The fundamental reason is that the routing path which is built based on trusted routing algorithms is trustworthy. The source node can transmit the data packet to destinations without passing a malicious node.

We can see from Figure 6(b), the impact of the number of malicious nodes on end-to-end delay. The end-to-end delay is slowly growing in MTAODV, LWT-MAODV and RBTM-MAODV as the number of malicious nodes increases, while the packet delivery time in MAODV is not affected by the number of malicious nodes. The reasons for that are as follows: (1) the greater the number of malicious nodes in the network, the more times the number of available paths can be disconnected; (2) each node in the three trust-based protocols periodically broadcasts control packets for sharing the trust information; (3) the available channels are congested by the packets, delay occurs.

Figure 6(c) shows the changing trend of the control overhead with the increasing number of malicious nodes. The greater the number of malicious nodes in the network, the more control packets need to be broadcast, increasing the control overhead. The calculation method of malicious nodes in MTAODV is more complicated than the other two protocols, and more control packets need to be sent in the network, which leads to the maximum overhead.

According to Test 1 and 2, we can conclude that our new protocol can effectively improve the transmission rate of data



(a) Packet delivery ratio

(b) End-to-end delay

(c) Control overhead

FIGURE 6: Performance with varying the number of malicious nodes.

packets at the expense of a slight increase in end-to-end delay and control overhead.

6. Conclusions and Future Work

Vehicle ad hoc networks are vulnerable to various attacks due to their inherent features. It is relatively easy for multiple malicious entities to bring down the whole network in several network services. In recent years, the problem of routing security has become a significant concern for researchers. The trust-based countermeasure is considered to be more acceptable as a promising approach. The trust computing plays an important role to initialize a trusted network system. In this paper, we carry out a detailed study of the various trust-based countermeasures. More specifically, we first abstract a novel trust model. In the proposed trust model, direct trust is calculated based on Bayesian theory and indirect trust is computed according to evaluation credibility and activity. We subsequently proposed an efficient trust-based multicast routing protocol (MTAODV) on the basis of standard MAODV protocol, which is used to defend against multiple attacks and improve the routing efficiency.

In future work, we plan to conduct an in-depth study of trusted routing strategies, taking into account the requirements for deployment area issues, network applications, and security levels [35]. Moreover, trust computations and management can be an attractive target for attackers, since major decisions can be taken based on these trust computations.

Hence, defence mechanisms are also needed to be designed at the same time [36].

Data Availability

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

Conflicts of Interest

The received funding does not lead to any conflicts of interest regarding the publication of this manuscript.

Acknowledgments

This work is sponsored by the Natural Science Foundation of China (NSFC) under Grant no. 61872205, the Project of Shandong Province Higher Educational Science and Technology Program no. J16LN06, Source Innovation Programme of Qingdao no. 18-2-2-56-jch, and the State Foundation of China for Studying Abroad to Visit the United States as a 'Visiting Scholar'.

References

[1] S. K. Khaitan and J. D. McCalley, "Design techniques and applications of cyberphysical systems: A survey," *IEEE Systems Journal*, vol. 9, no. 2, pp. 350–365, 2015.

[2] Y. Zhou, Z. Mo, Q. Xiao, S. Chen, and Y. Yin, "Privacy-Preserving Transportation Traffic Measurement in Intelligent Cyber-physical Road Systems," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3749–3759, 2016.

[3] Q. Yang, B. Zhu, and S. Wu, "An architecture of cloud-assisted information dissemination in vehicular networks," *IEEE Access*, vol. 4, pp. 2764–2770, 2016.

[4] S. Bitam, A. Mellouk, and S. Zeadally, "Bio-inspired routing algorithms survey for vehicular ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 843–867, 2015.

[5] J. Yu, K. Ren, and C. Wang, "Enabling cloud storage auditing with verifiable outsourcing of key updates," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1362–1375, 2016.

[6] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling Cloud Storage Auditing With Key-Exposure Resistance," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1167–1179, 2015.

[7] J. Yao, S. Feng, and X. Zhou, "Secure Routing in Multi-hop Wireless Ad-Hoc Networks With Decode-and-Forward Relaying," *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 753–764, 2016.

[8] W. Li and H. Song, "ART: an attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 960–969, 2016.

[9] Q. Yang and H. Wang, "Toward trustworthy vehicular social networks," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 42–47, 2015.

[10] M. Slavik and I. Mahgoub, "Spatial distribution and channel quality adaptive protocol for multihop wireless broadcast routing in VANET," *IEEE Transactions on Mobile Computing*, vol. 12, no. 4, pp. 722–734, 2013.

[11] G. Li, L. Boukhatem, and S. Martin, "An intersection-based QoS routing in vehicular ad hoc networks," *Mobile Networks and Applications*, vol. 20, no. 2, pp. 268–284, 2015.

[12] M. A. Togou, A. Hafid, and L. Khoukhi, "SCRP: Stable CDS-Based Routing Protocol for Urban Vehicular Ad Hoc Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1298–1307, 2016.

[13] K. Alodadi, A. H. Al-Bayatti, and N. Alalwan, "Cooperative volunteer protocol to detect non-line of sight nodes in vehicular ad hoc networks," *Vehicular Communications*, vol. 9, pp. 72–82, 2017.

[14] A. Mohamad Mezher and M. Aguilar Igartua, "Multimedia multimetric map-Aware routing protocol to send video-Reporting messages over VANETs in smart cities," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10611–10625, 2017.

[15] H. Fatemidokht and M. Kuchaki Rafsanjani, "F-Ant: an effective routing protocol for ant colony optimization based on fuzzy logic in vehicular ad hoc networks," *Neural Computing and Applications*, vol. 29, no. 11, pp. 1127–1137, 2018.

[16] F. Gómez Mármol and G. Martínez Pérez, "TRIP, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 934–941, 2012.

[17] X. Feng, C.-Y. Li, D.-X. Chen, and J. Tang, "A method for defensing against multi-source Sybil attacks in VANET," *Peer-to-Peer Networking and Applications*, vol. 10, no. 2, pp. 305–314, 2017.

[18] D. B. Rawat, G. Yan, B. B. Bista, and M. C. Weigle, "Trust on the security ofwireless vehicular Ad-hoc networking," *Ad-Hoc & Sensor Wireless Networks*, vol. 24, no. 3-4, pp. 283–305, 2015.

[19] D. Sun, H. Zhao, and S. Cheng, "A novel membership cloud model-based trust evaluation model for vehicular ad hoc network of T-CPS," *Security and Communication Networks*, vol. 9, no. 18, pp. 5710–5723, 2016.

[20] S. Dahmane, C. A. Kerrache, N. Lagraa, and P. Lorenz, "WeiSTARS: A Weighted Trust-Aware Relay Selection Scheme for VANET," in *Proceedings of the 2017 IEEE International Conference on Communications, ICC 2017*, 6, 1 pages, May 2017.

[21] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "Trust model for secure group leader-based communications in VANET," *Wireless Networks*, 2018.

[22] T. Bouali, E.-H. Aglzim, and S.-M. Senouci, "A Secure Intersection-Based Routing Protocol for Data Collection in Urban Vehicular Networks," in *Proceedings of the 2014 IEEE Global Communications Conference, GLOBECOM 2014*, pp. 82–87, December 2014.

[23] Q. Wu, Q. Liu, L. Zhang, and Z. Zhang, "A trusted routing protocol based on GeoDTN+Nav in VANET," *China Communications*, vol. 11, no. 2, pp. 166–174, 2014.

[24] C. A. Kerrache, N. Lagraa, C. T. Calafate, and A. Lakas, "TROUVE: A trusted routing protocol for urban vehicular environments," in *Proceedings of the 11th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2015*, pp. 260–267, 2015.

[25] T. Gazdar, A. Belghith, and H. Abutair, "An Enhanced Distributed Trust Computing Protocol for VANETs," *IEEE Access*, vol. 6, pp. 380–392, 2017.

[26] X. Yao, X. Zhang, H. Ning, and P. Li, "Using trust model to ensure reliable data acquisition in VANETs," *Ad Hoc Networks*, vol. 55, pp. 107–118, 2017.

[27] C. Hu, W. Li, X. Cheng, J. Yu, S. Wang, and R. Bie, "A Secure and Verifiable Access Control Scheme for Big Data Storage in Clouds," *IEEE Transactions on Big Data*, 2018.

[28] K. Xing, C. Hu, J. Yu, X. Cheng, and F. Zhang, "Mutual privacy preserving *k*-means clustering in social participatory sensing," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2066–2076, 2017.

[29] S.-J. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 441–453, 2002.

[30] Z. Cai, Z.-Z. Chen, and G. Lin, "A 3.4713-approximation algorithm for the capacitated multicast tree routing problem," *Theoretical Computer Science*, vol. 410, no. 52, pp. 5415–5424, 2009.

[31] Z. Cai, R. Goebel, and G. Lin, "Size-constrained tree partitioning: approximating the multicast *k*-tree routing problem," *Theoretical Computer Science*, vol. 412, no. 3, pp. 240–245, 2011.

[32] Z. Cai, Z.-Z. Chen, G. Lin, and L. Wang, "An improved approximation algorithm for the capacitated multicast tree routing problem," in *Combinatorial Optimization and Applications: Second International Conference, COCOA 2008, St. John's, NL, Canada, August 21–24, 2008. Proceedings*, vol. 5165 of *Lecture Notes in Computer Science*, pp. 286–295, Springer, Berlin, Germany, 2008.

[33] "A discrete event simulator ns-2," 2017, https://www.isi.edu/nsnam/ns/.

[34] F. Xiao, W. Liu, Z. Li, L. Chen, and R. Wang, "Noise-tolerant wireless sensor networks localization via multi-norms regularized matrix completion," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2409–2419, 2018.

[35] H. Xia, Z. Li, Y. Zheng, A. Liu, Y. C. Choi, and H. Sekiya, "A Novel Light-weight Subjective Trust Inference Framework in MANETs," *IEEE Transactions on Sustainable Computing*, 2018.

[36] H. Xia, F. Xiao, S. Zhang, X. Cheng, and Z. Pan, "A Reputation-Based Model for Trust Evaluation in Social Cyber-Physical Systems," in *IEEE Transactions on Network Science and Engineering*, 2018.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 7963163, 15 pages
https://doi.org/10.1155/2018/7963163

Research Article

Achieving the Optimal k-Anonymity for Content Privacy in Interactive Cyberphysical Systems

Jinbao Wang[1], **Ling Tian**[2], **Yan Huang**[3], **Donghua Yang**[1] and **Hong Gao**[4]

[1] *The Academy of Fundamental and Interdisciplinary Sciences, Harbin Institute of Technology, China*
[2] *School of Computer Science and Technology, University of Electronic Science and Technology of China, China*
[3] *Department of Computer Science, Georgia State University, USA*
[4] *School of Computer Science and Technology, Harbin Institute of Technology, China*

Correspondence should be addressed to Ling Tian; lingtian@uestc.edu.cn and Donghua Yang; yang.dh@hit.edu.cn

Received 14 June 2018; Revised 18 August 2018; Accepted 29 August 2018; Published 26 September 2018

Academic Editor: Liran Ma

Modern applications and services leveraged by interactive cyberphysical systems (CPS) are providing significant convenience to our daily life in various aspects at present. Clients submit their requests including query contents to CPS servers to enjoy diverse services such as health care, automatic driving, and location-based services. However, privacy concerns arise at the same time. Content privacy is recognized and a lot of efforts have been made in the literature of privacy preserving in interactive cyberphysical systems such as location-based services. Nevertheless, neither the cloaking based solutions nor existing client based solutions have achieved effective content privacy by optimizing proper content privacy metrics. In this paper we formulate the problem of achieving the optimal content privacy in interactive cyberphysical systems using k-anonymity solutions based on two content privacy metrics, which are defined using the concepts of entropy and differential privacy. Then we propose an algorithm, Multilayer Alignment (MLA), to establish k-anonymity mechanisms for preserving content privacy in interactive cyberphysical systems. Our proposed MLA is theoretically proved to achieve the optimal content privacy in terms of both the entropy based and the differential privacy mannered content privacy metrics. Evaluation based on real-life datasets is conducted, and the evaluation results validate the effectiveness of our proposed algorithm.

1. Introduction

Cyberphysical systems (CPS), which deeply integrate different computing, communication, controlling, and monitoring components, have leveraged modern services in our daily life, like smart grid, intelligent transportation, automatic driving, etc. Recent development of mobile communication and networks has leveraged many modern applications built on interactive cyberphysical systems, in which client software programs or devices take actions according to their interactions with CPS servers. In more details, a client sends a request to the CPS server and is to take actions on receiving the reply from the CPS server. The actions to be taken depend on the reply of CPS servers. Health caring, automatic driving, and location-based services fall into this category of interactive CPS applications. Suppose an old guy Bob is wearing a health caring device, which is connected to a CPS server though mobile Internet. Bob could send "stomachache" to the CPS server for instructions to help him. The CPS server may send a reply telling Bob what to do or where is the nearest hospital. Then Bob takes his action according to the reply. Similar processes could be adopted in applications of automatic driving and location-based services. These modern services over interactive CPS are attractive since they do bring convenience to people's daily life. However, privacy concerns arise at the same time while users' requests are submitted to the CPS servers through Internet. These requests disclose users' query contents to the CPS server and even vicious listeners to the communication channels. Here we refer query content as parameters in users' requests, such as "stomachache" in Bob's request to the health caring CPS server. These query contents should be kept as sensitive information for individuals, and the abuse or further leakage of these information will make users

vulnerable in respect of private life or even individual security [1].

To the privacy concern, content privacy should be recognized to emphasize that users' query contents should be kept as sensitive information in interactive cyberphysical systems. Many research efforts have been made to protect different types of query contents such as locations and keywords in the literature of interactive cyberphysical systems such as location-based services. The major body of these efforts consists of two parts, cloaking based solutions and client based solutions. Cloaking based solutions employ a trusted server from a third party. When a user u queries with a query content $u.content$, the query $q(u.content)$ is sent to the trusted server which in the next step generates a cloaking region CR containing u's location and at least another $k-1$ users. Here k specifies the level of privacy guarantee. Then the query with CR is sent to the CPS server, and the CPS server could not determine where u is or even whether u is querying. In this process, cloaking based solutions aim to make u indistinguishable from another $k-1$ users; however it suffers from inherent drawbacks brought by the trusted server, which may become the single point of failure of privacy and the bottleneck of query performance. More seriously, when the CPS server holds certain side information such as the prior probability of query contents, cloaking based solutions will suffer from further privacy breach. To address this issue, client based solutions are presented, aiming at k-anonymity provided by the client side. Reference [2] generates $k-1$ dummy query contents for continuous scenarios. To prevent the adversary from inferring the actual query content, [2] constrains that the selected dummies should have prior probability larger than a predefined threshold. In practice, it is difficult to determine a proper threshold, and what is more, dummies selected by [2] could still be eliminated if they have quite different prior probability. Reference [3] employs an entropy based privacy metric and generates $k-1$ dummy locations in random manner. However, the improper dummies could be eliminated from the reported k query contents due to the process of [3]; thus the provided privacy is degraded.

In this paper, we investigate the problem of preserving content privacy in interactive cyberphysical systems with a client based solution. To guarantee the utility of requests to the CPS servers, we adopt k-anonymity in order to prevent the adversary from recognizing the actual query content from the k reported contents, since the actual query content must be sent to the CPS server for a meaningful reply. In this process, the major challenge arises in two aspects. First, the k-anonymity provided should be carefully designed so that the adversary is not able to eliminate any query contents. Second, the overall privacy produced by k-anonymity should be optimized. We present two privacy metrics denoted expected entropy and dp-ratio which depict the achieved content privacy using a entropy based concept and a differential privacy mannered measurement, respectively. Then an algorithm, Multilayer Alignment (MLA), which establishes k-anonymity based mechanisms for preserving content privacy is proposed. Given the prior probability of query contents together with an integer k which specifies the privacy level, MLA generates a set of reports, each of which consists of k distinct query contents, together with probability distribution on the report set for each query content. Given any report r submitted to the CPS server, MLA guarantees that the posterior probability of each query content in r is larger than 0. To this end, the adversary is not able to eliminate any query content from a report. Here a report could be taken as a set of different query contents, and we give its formal definition in Section 3. We theoretically introduce the properties of MLA by proving that MLA achieves the optimal expected entropy and the optimal dp-ratio at the same time. These attractive properties make MLA the optimal k-anonymity solution for preserving content privacy in interactive cyberphysical systems. The major contributions of this paper are as follows.

(i) We formulate the problem of achieving the optimal k-anonymity based mechanisms for preserving content privacy in interactive cyberphysical systems. The problem formulation is based on two content privacy metrics with entropy and differential privacy concepts.

(ii) We propose the Multilayer Alignment (MLA) algorithm, which establishes k-anonymity based mechanisms for preserving content privacy. The MLA algorithm prevents adversaries from eliminating query contents from reports using Bayes inference.

(iii) We prove that MLA achieves the optimal k-anonymity mechanisms in terms of our presented content privacy metrics simultaneously.

(iv) We evaluate our proposed MLA algorithm using real-life datasets. The evaluation results validate that MLA achieves effective content privacy in terms of the entropy based and differential privacy mannered content privacy metrics.

The rest of this paper is organized as follows. Section 2 introduces some necessary preliminaries including the process of preserving content privacy using a client based solution, together with common accepted privacy metrics. Section 3 formulates the problem of achieving the optimal k-anonymity for content privacy in interactive cyberphysical systems. Section 4 proposes the MLA algorithm which establishes effective mechanisms for preserving content privacy. Section 5 theoretically proves that our proposed MLA algorithm achieves the optimal k-anonymity for content privacy in terms of the content privacy metrics introduced in Section 3. Section 6 evaluates the MLA algorithm based on real-life datasets and related work of this paper is discussed in Section 7. Finally, Section 8 concludes this paper.

2. Preliminary

This section introduces necessary preliminaries including the process of preserving content privacy using a client based k-anonymity solution. Then two accepted privacy

notions, i.e., k-anonymity and differential privacy, and their corresponding metrics are introduced.

2.1. Client Based k-Anonymity Solution. In the process of a client based k-anonymity solution for content privacy preservation, k query contents are reported to the CPS server when a user u wants to submit a request. The k reported query contents are determined on u's device, and no trusted third-party servers are employed; thus the potential single point of privacy failure and query performance bottleneck are eliminated. It is worth noticing that the actual query content u queries should be included in the reported ones; otherwise u is not able to receive a meaningful reply from the CPS server. After receiving the reported query contents, the CPS server processes k queries, one for each reported query content, and then returns the query results to u. Irrelevant results are filtered on u's mobile device and the actual results are returned to u. In this process, the CPS server receives k distinct query contents instead of a single actual one, and to this end the actual query content is hidden. Nevertheless, careful design is required to avoid ineffective dummies in the reported query contents. The way of generating reports to the CPS server determines the level of content privacy achieved, and this motivates our work.

2.2. Privacy Notions and Metrics

2.2.1. k-Anonymity. One widely adopted notion of privacy is k-anonymity, which is firstly introduced in the database community by [4]. The principle of k-anonymity is to hide the sensitive information into $k-1$ dummies so as to make the adversary unable to recognize the actual one. In the literature of privacy protection in interactive cyberphysical systems, k-anonymity could be categorized into cloaking based solutions and client based solutions. The cloaking based solutions such as [5] employ a third-party but trusted server, which is responsible for hiding the actual user among at least $k-1$ dummy users by spatial generalization. The client based solutions including more recent work such as [2, 3, 6] perform on users' devices and generate $k-1$ dummies in a local manner, in which process certain side information is adopted, for instance, the prior probability of each query content. The trusted server in cloaking based solutions may become a single failure if it is hacked by attackers and it is the performance bottleneck to incur long latency to requests. What is more, most of cloaking based solutions is unaware of side information held by the adversary such as the prior probability of query contents. At the same time, the existing client based solutions provide specious k-anonymity, since the attackers may violate the principle of k-anonymity through rerunning of the algorithms or launching probability inference for each of the k query contents.

The quality of k-anonymity could be measured by the concept of entropy borrowed from the area of information theory. When the CPS server receives a report $r = \{c_1, \ldots, c_k\}$ consisting of k query contents, the entropy of r is formulated as follows:

$$H(r) = -\sum_{i=1}^{k} Pr(c_i \mid r) \log Pr(c_i \mid r) \quad (1)$$

Note that the former formulation is slightly different from [3]. Actually, [3] takes the prior probability $Pr(c_i)$ as an approximation of the posterior probability $Pr(c_i \mid r)$.

2.2.2. Differential Privacy. Differential privacy was firstly introduced and applied in statistic databases, and it aims to prevent the leakage of any individual's information during query processing. Generally speaking, to satisfy the notion of differential privacy, a random algorithm should return query results with similarly distribution for two databases differing with just one tuple. In other words, a single modification in a database brings a minor change to query results under the control of differential privacy. The definition of differential privacy is given below.

Definition 1 (differential privacy). Given $\epsilon \geqslant 0$, a randomized algorithm $\mathscr{A}$ satisfies ϵ-differential privacy if for all neighboring databases D and D', $Prob(\mathscr{A}(D) \in S) \leqslant e^{\epsilon} \times Prob(\mathscr{A}(D') \in S)$. Here $S \subseteq Range(\mathscr{A})$. Any pair of neighboring databases D and D' satisfies one of the following conditions: (1) (for unbounded differential privacy) D can be transformed to D' with exact one insertion or deletion; (2) (for bounded differential privacy) D can be transformed to D' with exact one modification.

The bounded differential privacy prevents distinguishing two datasets with the same size while differing with exact one tuple. The unbounded differential privacy prevents distinguishing two datasets which are the same except that one of them holds exact one additional tuple.

The metric for differential privacy is the coefficient ϵ in Definition 1. Intuitively, a smaller ϵ leads to a better privacy but larger noise in the query result, while a larger ϵ leads to less noise in the query result but a weaker privacy guarantee.

3. Problem Definition

This section formulates the problem of achieving the optimal k-anonymity based mechanisms for content privacy in interactive cyberphysical systems. Before introducing the problem definition, we provide several definitions which interpret indispensable concepts for our problem definition.

When a user queries a content c, the client based solution first generates a report consisting of k distinct query contents and then sends the report to the CPS server for response. A formal definition of a report is given as follows.

Definition 2 (report). Given the global set C of query contents and an integer $k > 0$, a report r is a subset of C with size k. When a user queries content c, the generated report r must contain c; i.e., $c \in r$. Denote the set of all the reports for the given C and k by R_C^k. The set of reports containing the query content c is denoted by $R_c^{C,k}$.

In the rest of this paper, we focus on specified C and k, and we also use the notion R_c (instead of $R_c^{C,k}$) to refer to the set of reports containing the query content c.

For a client based solution, multiple reports could include an identical query content c. When c is queried, one of these reports is submitted to the CPS server. The following definition of reporting probability depicts the process of selecting such a report.

Definition 3 (reporting probability). Given the global set C of query contents and an integer $k > 0$, a reporting probability is a function $Pr_C^k : C \times R_C^k \longrightarrow [0, 1]$ satisfying the following constraints:

(i) for any $c \in C$ and $r \in R_c$, $Pr_C^k(c, r) \geq 0$;

(ii) for any $c \in C$, $\sum_{r \in R_c} Pr_C^k(c, r) = 1$;

(iii) for any $c \in C$ and $r \notin R_c$, $Pr_C^k(c, r) = 0$.

The first two constraints in the definition of reporting probability illustrate that when querying a content $c \in C$, a report $r \in R_c$ is selected according to the probability $Pr_C^k(c, r)$. The third constraint specifies that a report r will not be selected for c if $c \notin r$.

Next we formulate a client based solution as a mechanism in a probabilistic manner based on the concepts of report and reporting probability.

Definition 4 (mechanism). Given the global set C of the query contents and an integer $k > 0$, a k-anonymity based mechanism $\mathscr{M}$ consists of two components including the set of reports R_C^k and the reporting probability Pr_C^k. When a user queries content $c \in C$, $\mathscr{M}$ randomly selects a report r from R_C^k and the probability of selecting r is $Pr_C^k(c, r)$.

The above definition of a k-anonymity based mechanism looks speciously strange; nevertheless existing solutions could be taken as instances of the above definition. We could specify the reporting probability using additional parameters in these works, for instance, the predefined prior probability threshold σ in [2].

This paper adopts two privacy metrics to measure a given k-anonymity based mechanism $\mathscr{M}$ in terms of privacy. As formulated in the following definition, the first metric integrates entropy measures of all the reports generated by $\mathscr{M}$.

Definition 5 (expected entropy). Given the global set C of query contents, an integer $k > 0$ and the prior probability of query contents as $pr(.)$, the expected entropy of mechanism $\mathscr{M}(R_C^k, Pr_C^k)$ is calculated as

$$\begin{aligned} H_E(\mathscr{M}) &= \sum_{r \in R_C^k} Pr(r) H(r) \\ &= \sum_{r \in R_C^k} \sum_{c \in r} pr(c) Pr_C^k(c, r) \sum_{c \in r} -Pr(c \mid r) \log Pr(c \mid r). \end{aligned} \tag{2}$$

Here $Pr(c \mid r) = pr(c) Pr_C^k(c, r) / \sum_{i \in r} pr(i) Pr_C^k(i, r)$, and it is the posterior probability of c given report r. $H_E(\mathscr{M})$ measures the achieved content privacy overall by considering all the generated reports. The probability of each report is taken as the weight, and the entropy of each report is integrated in the above formulation. A larger $H_E(\mathscr{M})$ indicates that a better content privacy is obtained with respect to the concept of entropy.

The second metric incorporates the notion of k-anonymity and differential privacy. It measures a mechanism with the most distinguishable pairs of query contents in the generated reports. The following definition formulates our second metric named dp-coefficient.

Definition 6 (dp-coefficient). Given the global set C of query contents, an integer $k > 0$, and the prior probability of query contents as $pr(.)$, the dp-coefficient of mechanism $\mathscr{M}(R_C^k, Pr_C^k)$ is calculated as

$$dpc(\mathscr{M}) = \max_{r \in R_C^k, i, j \in r} \ln \frac{Pr(i \mid r)}{Pr(j \mid r)}. \tag{3}$$

Here the terms $Pr(i \mid r)$ and $Pr(j \mid r)$ are the posterior probability of query contents i and j given a report r, and they could be calculated in the same way as the calculation of $Pr(c \mid r)$ described above.

Based on the content privacy metrics, e.g., expected entropy and differential privacy coefficient, we formulate the problem of achieving the optimal k-anonymity for content privacy in interactive cyberphysical systems as follows.

Problem Definition. Given the global set C of query contents, an integer $k > 0$, and the prior probability of query contents as $pr(.)$, compute a mechanism $\mathscr{M}(R_C^k, Pr_C^k)$ with the optimal content privacy. The optimal content privacy is achieved if $H_E(\mathscr{M})$ is maximized.

$H_E(\mathscr{M})$ and $dpc(\mathscr{M})$ depict the content privacy achieved by $\mathscr{M}$ from the holistic and individual point of view, respectively. Although our problem definition aims at the optimized expected entropy, in the next section we propose an algorithm which achieves the optimal expected entropy and the optimal dp-coefficient simultaneously.

4. Achieving the Optimal k-Anonymity

This section in first provides a short discuss on a naïve solution to the problem defined in Section 3. Then we propose our Multilayer Alignment (MLA) algorithm which achieves the optimal k-anonymity for content privacy in interactive cyberphysical systems. MLA exhibits an attractive property that it achieves the maximized expected entropy and the minimized dp-coefficient simultaneously.

4.1. A Naïve Solution. According to the problem definition in Section 3, the essential challenge of establishing the optimal mechanisms lies in building the reporting probability $Pr_C^k : C \times R_C^k \longrightarrow [0, 1]$. A naïve approach to achieve the optimal k-anonymity is formulating the problem using nonlinear-programming technique with linear constraints in Definition 3, and expected entropy or dp-coefficient is used as the

optimizing objective. However, the nonlinear-programming formulation employs $\binom{|C|}{k} \times |C|$ variables each of which stands for an entry in Pr_C^k. When $|C|$ grows to 100 and k is set to 10, there will be more than 10^{15} variables. Thus this naïve approach is impractical due to its computation expense.

4.2. The Multilayer Alignment Algorithm. MLA computes the optimal k-anonymity mechanism in two phases, namely, (1) Segment Alignment and (2) Mechanism Initiation. The major idea of MLA is to generate a mechanism where query contents have as similar posterior probability as possible in each report. To accomplish this goal in a holistic manner, Segment Alignment amortizes each query content with large prior probability to multiple query contents with small prior probability. What is more, the reports generated by Mechanism Initiation have the same distribution of posterior probability of the k included query contents. Next we introduce the two phases of MLA.

4.2.1. Segment Alignment. Given the prior probability $pr(.)$ of query contents, MLA represents each query content $c_i \in C$ using a segment s_i with length $|s_i| = pr(c_i)$. The segments for all the query contents are sorted in descend order, and denote the sorted set as $Seg = \{s_1, \ldots, s_m\}$. Then MLA aligns the segments onto k layers in order. The aligning process has two modes, i.e., aligning dominant and aligning dominated. At the beginning of aligning, the mode of aligning dominant is active. The number of rest layers (denoted $rest_{layer}$) is set to k. MLA checks whether the current segment is dominant. When aligning s_i, s_i is dominant if the condition $|s_i| \times rest_{layer} > \sum_{i \geq j \geq m} |s_j|$ holds. If the current segment s_i is dominant, MLA aligns s_i onto the current layer and s_i takes up the entire layer. The aligning stays in mode aligning dominant, and segment s_{i+1} is taken as the current segment when the aligning continues. If the current segment s_i is not dominant, the aligning turns to mode aligning dominated. Then MLA sets the length of each of the remaining layers as $\sum_{i \geq j \geq m} |s_j| / rest_{layer}$, and it aligns $s_i, \ldots, s_m$ along the rest layers. When aligning a segment s_j and the current layer has blank length l_b less than $|s_j|$, s_j is divided into two parts with lengths l_b and $|s_j| - l_b$. The first part is aligned onto the current layer, and the second part is aligned onto the beginning of the next layer. In the mode of aligning dominated, MLA goes on aligning all the remaining segments, and it never turns back to mode aligning dominant. After all the remaining segments are aligned, the first phase of MLA terminates and MLA continues to the second phase.

Example 7. Suppose $C = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$ and Alice wants to query c_1 nearby, and the prior probability of each query content at her location is given as follows: $pr(c_1) = 0.4$, $pr(c_2) = 0.15$, $pr(c_3) = 0.15$, $pr(c_4) = 0.1$, $pr(c_5) = 0.1$, $pr(c_6) = 0.05$, and $pr(c_7) = 0.05$. Alice desires for 4-anonymity ($k = 4$), and what is the optimal mechanism for Alice?

Here we use the instance in Example 7 to illustrate the process of segment aligning. There are 4 layers in the process of aligning. The segment for c_1 is aligned in mode aligning dominant since $pr(c_1) \times k > 1$. The segment for the remaining query contents is aligned in mode aligning dominated, and each of the remaining layers is at length $(1 - 0.4)/(4 - 1) = 0.2$. The aligning result is shown in Figure 1.

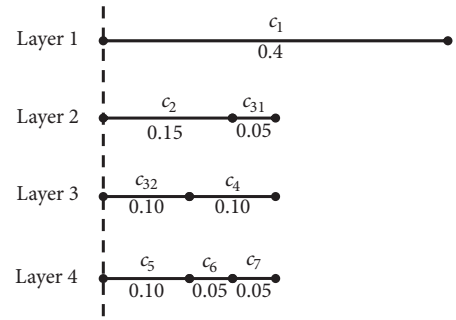


Figure 1: Segment aligning for the instance consisting of 7 query contents in Example 7. The first layer is taken up by c_1 under aligning dominant mode, and the remaining 3 layers are taken up by c_2, c_3, c_4, c_5, c_6, and c_7, under aligning dominated mode. Here c_{ij} means the jth part of content i.

4.2.2. Mechanism Initiation. Denote the length of the ith layer by L_i, where $i = 1, \ldots, k$. The second phase of MLA first shrinks the length of each layer, and the ith layer is shrunk by ratio L_i/L_k. The shrinking ratio of the ith layer is recorded as $ratio_i$. All the layers have the same length after shrinking. In the next step, MLA sets a vertical line at the beginning of each layer. Then the vertical line moves to the right until it touches the first point on any layer at which a segment ends. Then the scanned parts of the k layers are packed into a report. Denote the scanned part on the ith layer with length l_i^s; then the probability of this report is $\sum_{1 \geq i \geq k} l_i^s \times ratio_i$ and the posterior probability of the query content on the ith layer is $(l_i^s \times ratio_i)/(\sum_{1 \geq j \geq k} l_j^s \times ratio_j)$. The vertical line continues moving to the right and MLA packs the next report when any segment ends on a layer. The process terminates after the vertical line moves to the end of each layer and generates the last report. Continue with Example 7 as shown in Figure 2, the shrinking ratios of the 4 layers are 2, 1, 1 and 1. Then a vertical line starts moving to the right from the left end of all the layers. It first touches the end points of c_{32} and c_5 on layers 3 and 4, respectively, and a report $r_1 = \{c_1, c_2, c_3, c_5\}$ is generated. Then it keeps moving to the right and touches the end points of c_2 and c_6 on layers 2 and 4, respectively, and report $r_2 = \{c_1, c_2, c_4, c_6\}$ is generated. Finally, the vertical line touches the end points of all the layers and generates the last report $r_3 = \{c_1, c_3, c_4, c_7\}$. In the end, MLA generates 3 reports including $r_1 = \{c_1, c_2, c_3, c_5\}$, $r_2 = \{c_1, c_2, c_4, c_6\}$, and $r_3 = \{c_1, c_3, c_4, c_7\}$. The reporting probability is given in Table 1. Take c_1, for instance; half of its prior probability is assigned into report r_1, and one-fourth of its prior probability is assigned to reports r_2 and r_3. Thus the reporting probability of c_1 for reports r_1, r_2, and r_3 is 1/2, 1/4, and 1/4, respectively, as shown in Table 1. The reporting probability of other contents could be calculated in the same manner.

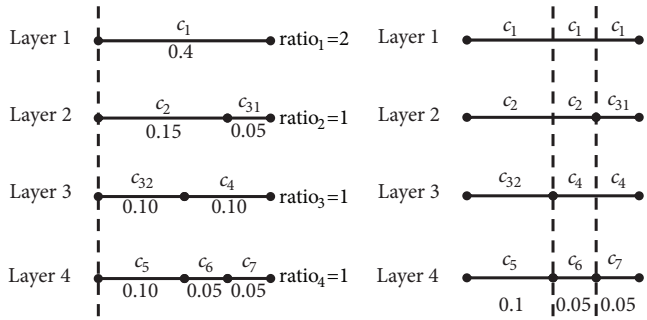


FIGURE 2: The first layer is shrunk by ratio of 2, and remaining layers keep unchanged. Mechanism initiation produces three reports $r_1 = \{c_1, c_2, c_3, c_5\}$, $r_2 = \{c_1, c_2, c_4, c_6\}$, and $r_3 = \{c_1, c_3, c_4, c_7\}$. When querying c_1, the mechanism adopts reporting distribution $Pr(r_1 \mid c_1) = 0.5$, $Pr(r_2 \mid c_1) = 0.25$ and $Pr(r_3 \mid c_1) = 0.25$.

TABLE 1: The reporting probability for Example 7.

Content / Report	r_1	r_2	r_3
c_1	1/2	1/4	1/4
c_2	2/3	1/3	0
c_3	2/3	0	1/3
c_4	0	1/2	1/2
c_5	1	0	0
c_6	0	1	0
c_7	0	0	1

The pseudocode of the Multilayer Alignment algorithm is shown in Algorithm 1. In the beginning, MLA sorts the query contents in C according to their prior probability in descending order (line 1). Then it initiates necessary structures and variables. $L_1, \ldots, L_k$ stores the alignment of k layers (lines 2-3). Variable *lcusor* indicates the layer being processed (line 4), and *isDominant* indicates whether the alignment is under aligning dominant mode (line 5). Variable *partial* indicates how much prior probability of the current query content is taken up by the last layer while *average* indicates the length of each layer processed in dominated mode, and they are initiated in line 6. Array *Len* keeping the lengths of k layers is initiated in line 7. The loop in lines 8-28 aligns query contents in C in order onto k layers. The aligning mode is set dominant in line 5 before processing the first query content. Under aligning dominant mode, MLA checks whether c_i should be aligned under aligning dominant mode. If the answer is yes, a segment for c_i is created with length $pr(c_i)$ and it is added to the list L_{lcusor} for the current layer. Here the constructor of segment specifies the label and the length for a segment. Then the alignment of c_i and the current layer terminates (lines 10-12). If c_i should not be aligned under aligning dominant mode, MLA turns to the mode of aligning dominated and calculates the length of each of the remaining layers as *average* (lines 13-15). In aligning dominated mode, MLA executes the code in lines 16-28. If c_i could be entirely aligned onto the current layer (line 16), MLA creates a segment for c_i with length $pr(c_i)$, and adds it to list L_{lcusor} (line 17). The length of the current layer and *partial* are updated (lines 18-19). If the current layer does not have sufficient space to hold c_i, c_i is split into two segments. Lines 21-23 align the first segment onto the current layer, and lines 26-28 align the second segment onto the next layer. Lines 24-25 deal with a special case where c_i exactly uses up the space of the current layer. By here the Segment Alignment terminates and the MLA goes to the phase of Mechanism Initiation. It packs the heads of k lists L_i into a report r (lines 30-31). Then MLA determines the movement length ratio of the vertical line to the right as *step* (line 32). The reporting probability related to the current report is calculated in line 35. For each layer, MLA updates the length of the head. If the head of a layer is entirely packed into a report, it is popped from the list (lines 36-39).

The computation cost of MLA consists of three parts, sorting C and initiating variables, Segment Alignment, and Mechanism Initiation. The first part costs $O(|C| \log |C|)$. Segment Alignment costs $O(k + |C|)$ since at most $k + |C|$ segments are aligned, and each alignment costs a constant time. Mechanism Initiation costs $O(k(k + |C|))$ which is dominated by packing at most $k + |C|$ reports (each packing costs $O(k)$) and calculating at most $k(k + |C|)$ entries for Pr_C^k. In practice, k should be set smaller than $|C|$. The total cost of MLA is $O(|C| \log |C| + k|C|)$.

5. Properties of MLA Algorithm

This section formally proves that MLA achieves the optimal expected entropy and the optimal dp-coefficient simultaneously. We first introduce some concepts which build necessary foundation for our formal proof.

Definition 8 (dominant content). Given the global set C of query contents, an integer $k > 0$, and the prior probability $pr(.)$, $\forall c \in C$, let dom(c) be the number of query contents larger than c. A query content $c \in C$ is a dominant content iff the following conditions hold:

(i) $\text{dom}(c) < k$;

(ii) $\sum_{pr(c') \le pr(c)} pr(c') \le pr(c)(k - \text{dom}(c))$.

Input: query content set C, integer $k < |I|$, the prior probability $pr(.)$ of query content
Output: the reporting probability Pr_C^k

sort C according to $pr(.)$ in descend order;
for $1 \le i \le k$ **do**
    initiate a list L_i;
$lcusor = 1$;
$isDominant = true$;
$partial, average = 0$;
initiate an integer array Len with k values of 0;
for $1 \le i \le |C|$ **do**
    if $isDominant$ **then**
        if $pr(c_i) \ge sum_{pr}/(k + 1 - lcusor)$ **then**
            $L_{lcusor++}.add$(new Segment($i, pr(c_i)$));
            continue;
        else
            $isDominant = false$;
            $average = \sum_{i \le j \le |C|} pr(c_j)/(k + 1 - lcusor)$;
    if $pr(c_i) - partial < average - Len[lcusor]$ **then**
        $L_{lcusor}.add$(new Segment($i, pr(c_i) - partial$));
        $Len[lcusor] = Len[lcusor] + pr(c_i)$;
        $partial = 0$;
    else
        seg = new Segment($i, average - Len[lcusor]$);
        $L_{lcusor}.add(seg)$;
        $Len[lcusor++] = average$;
        if $seg.length == pr(c_i)$ **then**
            continue;
        seg' = new Segment($i, pr(c_i) - seg.length$);
        $L_{lcusor}.add(seg')$;
        $Len[lcusor] = seg'.length$;
while $L_1 \ne \varnothing$ **do**
    initiate a report r;
    add the heads of $L_1, \ldots, L_k$ to r;
    $step = \min_{1 \le i \le k} (L_i.head.length/Len[i])$;
    for $1 \le i \le k$ **do**
        $j = L_i.head.label$;
        $Pr_C^k(c_j, r) = (step \times Len[i])/pr(c_j)$;
        if $L_i.head.length == step \times Len[i]$ **then**
            $L_i.pop()$;
        else
            $L_i.head.length -= step \times Len[i]$;
return Pr_C^k;

Algorithm 1: Multilayer Alignment.

Definition 9 (dominated content). Given the global set C of query contents, an integer $k > 0$, and the prior probability $pr(.)$, $c \in C$ is a dominated content, not a dominant content.

According to the process of segment alignment in MLA, each dominant content takes up an entire layer. If there are remaining layers, the dominated contents take up these layers and none of them take up an entire layer. In the rest of this paper, we use dominant layer and dominated layer to denote a layer taken up by a dominant content or dominated contents, respectively. Recall the instance in Figure 1; c_1 is a dominant content and layer 1 is a dominant layer. Query contents $c_2, \ldots, c_7$ are dominated contents and layers 2, 3, and 4 are dominated layers.

Definition 10 (layering strategy). Given the global set C of the query contents, an integer $k > 0$, the prior probability $pr(.)$ of query contents, and a reporting probability Pr_C^k, let the query contents in each report be permuted arbitrarily and denote the ith query content of report r by c_i^r. A layering strategy induced by $\mathscr{M}$ is a k dimensional vector, whose ith component is calculated as $\sum_{r \in R_C^k} Pr_C^k(c_i^r, r) pr(c_i^r)$. When query contents are sorted by the value of $Pr_C^k(c, r) pr(c)$ in descend order, the standard layering strategy is induced.

According to the above definition of layering strategy, a mechanism has multiple layering strategies. Intuitively, a mechanism assigns the prior probability of each query

content to one or multiple reports. A report r contains k parts from distinct query contents, and they can be viewed as k segments on k layers. When query contents in each report are permuted, we can build k layers by connecting all the segments on the same layer from different reports together. To this end, we call the k dimensional vector a layering strategy. Next we define the entropy of a layering strategy.

Definition 11 (entropy of layering strategy). Given a layering strategy $ls = \{v_1, \ldots, v_k\}$, the entropy of ls is calculated as $H(ls) = -\sum_{1\le i\le k} v_i \log v_i$.

Lemma 12. *Given a mechanism $\mathscr{M}(R_C^k, Pr_C^k)$, let ls be any induced layering strategy of $\mathscr{M}$; then $H_E(\mathscr{M}) \le H(ls)$.*

Proof. Suppose the query contents in each report of M are arbitrarily sorted, and we get an induced layering strategy $ls = \{v_1, \ldots, v_k\}$. Denote the set of reports with posterior probability larger than 0 by $R = \{r_1, \ldots, r_m\}$, and $c_{i,j}$ is the jth query content of report r_i in the process of inducing ls. For $1 \le j \le k$, we have the following equation:

$$\begin{aligned} v_j &= \sum_{1\le i\le m} Pr\left(c_{i,j}\right) Pr\left(r_i \mid c_{i,j}\right) \\ &= \sum_{1\le i\le m} Pr\left(c_{i,j} r_i\right). \\ \sum_{1\le i\le m} Pr\left(r_i\right) &= 1. \end{aligned} \tag{4}$$

By applying the log-sum inequality [7] (adopted in the last but one line in the below), we have the following condition:

$$\begin{aligned} H_E(M) &= \sum_{1\le i\le m} Pr\left(r_i\right) H\left(r_i\right) \\ &= -\sum_{1\le i\le m} Pr\left(r_i\right) \sum_{1\le j\le k} Pr\left(c_{i,j} \mid r_i\right) \log Pr\left(c_{i,j} \mid r_i\right) \\ &= -\sum_{1\le i\le m} \sum_{1\le j\le k} Pr\left(c_{i,j} r_i\right) \log \frac{Pr\left(c_{i,j} r_i\right)}{Pr\left(r_i\right)} \\ &= -\sum_{1\le j\le k} \sum_{1\le i\le m} Pr\left(c_{i,j} r_i\right) \log \frac{Pr\left(c_{i,j} r_i\right)}{Pr\left(r_i\right)} \\ &\le \sum_{1\le j\le k} \left(\sum_{1\le i\le m} Pr\left(c_{i,j} r_i\right) \right) \left(\log \frac{\sum_{1\le i\le m} Pr\left(c_{i,j} r_i\right)}{\sum_{1\le i\le m} Pr\left(r_i\right)} \right) \\ &= -\sum_{1\le j\le k} v_j \log v_j = H\left(ls\right) \end{aligned} \tag{5}$$

So we prove that $H_E(M) \le H(ls)$. □

Lemma 13. *Given a mechanism $\mathscr{M}(R_C^k, Pr_C^k)$ generated by* MLA*, ls is the standard layering strategy of $\mathscr{M}$ and $\mathscr{M}'$ is an arbitrary mechanism; then $\mathscr{M}'$ has at least one induced layering strategy ls' satisfying the fact that $H(ls) \ge H(ls')$.*

Proof. Given the mechanism $\mathscr{M}$ produced by MLA together with its standard layering strategy ls, we prove Lemma 13 by conducting an induced layering strategy ls' for an arbitrary mechanism $\mathscr{M}'$, so that $H(ls) \ge H(ls')$. To this end, we sort the query contents in each report of $\mathscr{M}'$ as follows.

For each report r of $\mathscr{M}'$, we iterate all the query contents. For a query content c, if it is a dominating query content determined by MLA and its order in ls is $o_{ls}(c)$, we set the order of c in r by $o_{ls}(c)$. After arranging all the dominating query contents, we sort the rest of query contents in r by $pr(c) \times Pr(r \mid c)$ in descend order and then fill the blanks in the ordering of r. In this way we conduct an inducing layering strategy ls' of $\mathscr{M}'$, and in the following we are to prove that $H(ls) \ge H(ls')$.

Let n_{dom} be the number of dominant layers in ls, and we first investigate the first n_{dom} layers of ls'. For each dominant content on the jth layer of ls, it is also aligned only on the jth layer of ls'. At the same time, on the jth layer of ls' there are possibly dominated contents. So we get that for each dominant layer the length of ls' is no smaller than that of ls, i.e., $ls'.v_j \ge ls.v_j$, $1 \le j \le n_{dom}$. As a consequence, the total length of dominated layers in ls' is no larger than that of ls if $n_{dom} < k$; i.e., $\sum_{n_{dom}\le j\le k} ls'.v_j \le \sum_{n_{dom}\le j\le k} ls.v_j$.

Here we turn to a necessary observation of modifying a layering strategy at two layers with increased entropy. Suppose ls'' is an arbitrary layering strategy, and its values on layer j and layer h are different. With no loss of generality, assume $ls''.v_j > ls''.v_h$. Then we move a length of l from layer j to layer h; here $0 < l < 2(ls''.v_j - ls''.v_h)$. It is easy to see that the entropy of the modified layering strategy is larger than the entropy of ls''. Next we transform ls' to ls with a series of modifications of the above type between two layers with different lengths.

The transform includes two phases. In the first phase, ls' make the dominated layers (here the dominated layers and dominant layers are determined by ls) have the same length. Let l_{avg}^{dtd} be the average length of dominated layers for ls'. We repeat the below modification. Each time we pick the dominated layer with smallest length and largest length, and move the length from longer to the shorter until either one of them reaches l_{avg}^{dtd}. Then the number of layers with length l_{avg}^{dtd} increases by at least one. After at most $k - n_{dom}$ modifications, phase 1 terminates. And each modification make the entropy of ls' increase. If the dominated layers of ls' have the same length at the beginning of phase 1, its entropy remains unchanged.

In phase 2, we investigate each of the first n_{dom} layers. For a layer $j \le n_{dom}$, let the jth dominant content of MLA be c_j^{dom}. Then we remove a length of $ls'.v_j - pr(c_j^{dom})$ and distribute it evenly to dominated layers. After that, the length of each dominated layers for ls' is no larger than that of ls (denoted l_{dtd}). Meanwhile, the remaining length $pr(c_j^{dom})$ of layer j is larger than l_{dtd}. According to the observation above, each modification of phase 2 will increase the entropy of ls'. After at most n_{dom} modifications, ls' will be transformed to ls, and each modification will not decrease the entropy.

Combining phase 1 and phase 2, we conduct a transformation from an induced layering strategy ls' of an arbitrary mechanism to ls, which is an induced layering strategy of the mechanism produced by MLA. Each step of the conducted transformation will increase the entropy or keep the entropy unchanged, so we prove that $H(ls') \leq H(ls)$. □

Lemma 14. *Given a mechanism $\mathscr{M}(R_C^k, Pr_C^k)$ generated by* MLA*, and ls is the standard layering strategy of $\mathscr{M}$, then $H_E(\mathscr{M}) = H(ls)$.*

Proof. The standard layering strategy of $\mathscr{M}$ restores the result of segment alignment in the process of MLA. Let $L_1, \ldots L_k$ be the lengths of the k generated layers. Due to the shrinking process of MLA, the initiated reports have the same ratios between pairs of corresponding query contents on two given layers. Thus the standard layering strategy ls of $\mathscr{M}$ could be calculated as $ls = \{L_1/\sum_{1\leq j\leq k} L_j, L_2/\sum_{1\leq j\leq k} L_j, \ldots L_k/\sum_{1\leq j\leq k} L_j\}$. For each produced report r_i in the process of inducing ls, we use $c_{i,j}$ to denote the jth query content in r_i. Then we have $Pr(c_{i,j})Pr(r \mid c_{i,j})/Pr(c_{i,j'})Pr(r \mid c_{i,j'}) = L_j/L'_j$, for $1 \leq j, j' \leq k$. So we can get that the entropy of each report r_i equals the entropy of ls. As a consequence the expected entropy of $\mathscr{M}$ can be calculated as follows:

$$H_E(\mathscr{M}) = \sum_{r\in\mathscr{M}.R} Pr(r) H(r) = \sum_{r\in\mathscr{M}.R} Pr(r) H(ls) = H(ls). \quad (6)$$

So we prove that $H_E(\mathscr{M}) = H(ls)$. □

Lemmas 12, 13, and 14 illustrate the relationship between the expected entropy achieved by MLA and the entropy of induced layering strategies of any other mechanisms. Based on these facts, we get the following theorem.

Theorem 15. MLA *achieves the optimal expected entropy.*

Proof. According to Lemma 14, the mechanism $\mathscr{M}$ produced by MLA achieves the expected entropy of $H(ls)$, where ls is the standard induced layering strategy of $\mathscr{M}$. Assume $\mathscr{M}'$ is an arbitrary mechanism, and it has at least on induced layering strategy ls' so that $H(ls') \leq H(ls)$ due to Lemma 13. At the same time, $H_E(\mathscr{M}') \leq H(ls')$ according to Lemma 12. Then we have $H_E(\mathscr{M}) \geq H_E(\mathscr{M}')$, so we prove that MLA achieves the optimal expected entropy through $\mathscr{M}$. □

Theorem 16. MLA *achieves the optimal dp-coefficient.*

Proof. Given the mechanism $\mathscr{M}$ produced by MLA together with its standard layering strategy ls, we conduct an induced layering strategy ls' for an arbitrary mechanism $\mathscr{M}'$ in the same way as the proof of Lemma 13. We sort the query contents in each report of $\mathscr{M}'$ as follows. For each report r of $\mathscr{M}'$, we traverse its query contents. For a query content c, if it is a dominated content determined by MLA and its order in ls is $o_{ls}(c)$, we set the order of c in r by $o_{ls}(c)$. After arranging all the dominating query contents, we sort the rest of query contents in r by $pr(c) \times Pr(r \mid c)$ in descend order, and then fill the blanks in the ordering of r. The first layer of ls only contains the first dominant content; however the first layer of ls not only contains the first dominant content entirely but also possibly dominated contents. So we have $ls.v_1 \leq ls'.v_1$. On the other hand, we know that the total length of dominated layers in ls is no smaller than that of ls'. At the same time, each dominated layer has the same length in ls while the kth layer in ls' has the smallest length. Then we have $ls.v_k \geq ls'.v_k$. In $\mathscr{M}$ the dp-coefficient is actually $\ln(ls.v_1/ls.v_k)$. Denote the set of reports produced by $\mathscr{M}'$ by $\mathscr{M}'.R_C^k = \{r_1, \ldots, r_m\}$, and let $c_{i,j}$ be the jth query content in report r_i, $1 \leq i \leq m$ and $1 \leq j \leq k$. Then we have $\sum_{1\leq i\leq m} Pr(c_{i,1})Pr(r_i \mid c_{i,1}) = ls'.v_1$ and $\sum_{1\leq i\leq m} Pr(c_{i,k})Pr(r_i \mid c_{i,k}) = ls'.v_k$. Thus $\max_{1\leq i\leq m}(Pr(c_{i,1} \mid r)/Pr(c_{i,k} \mid r)) = Pr(c_{i,1})Pr(r \mid c_{i,1})/Pr(c_{i,k})Pr(r \mid c_{i,k}) \geq ls'.v_1/ls'.v_k$. The dp-coefficient achieved by $\mathscr{M}'$ is $\ln(Pr(c_{i,1} \mid r)/Pr(c_{i,k} \mid r)) \geq \ln(ls'.v_1/ls'.v_k)$. Since we have got that $ls.v_1 \leq ls'.v_1$ and $ls.v_k \geq ls'.v_k$, we conclude that $\ln(ls.v_1/ls.v_k) \leq \ln(ls'.v_1/ls'.v_k) \leq \ln(Pr(c_{i,1} \mid r)/Pr(c_{i,k} \mid r))$. That is to say the dp-coefficient of $\mathscr{M}$ is no larger than that of an arbitrary mechanism $\mathscr{M}'$. So we prove that MLA achieves the optimal dp-coefficient through $\mathscr{M}$. □

6. Evaluation

This section evaluates the performance of our proposed MLA algorithm based on three real-life datasets, and evaluation results report the comparison between MLA and three existing approaches including *MEE*, *MER* [8], and *DLS* [3].

6.1. Evaluation Setting

Datasets. To obtain the prior probability of query contents, we employ three real-life datasets including *TX* and *CA* from [9] and *POIs* from [10]. *TX* and *CA* contains street objects in the state of Texas and California. Each object is labeled with a coordinate and a set of keywords. *POIs* contains worldwide coordinates and geotags. We use the coordinates as locations, and take the keywords and geotags as query contents. We divide *TX*, *CA*, and *POIs* into $8km \times 8km$ regions and calculate a prior distribution of query contents for each region. Given the number of query contents $|C|$, we pick query contents with top-$|C|$ frequency and they are used to compute the prior probability. In *TX* and *CA*, some keywords such as city name and state name are removed since they dominate the frequency but provide no meanings. For each dataset, the average measures of its regions are reported in the evaluation results. The details of *TX*, *CA*, and *POIs* are introduced in Table 2.

Testbed. We implement our proposed MLA and competitors including *MEE*, *MER*, and *DLS* in Java language. The JDK version is jdk1.8.0_151. All of the evaluation is conducted on

Table 2: Details of datasets in evaluation.

Name	Description	Original/Preprocessed Size	Number of Words
TX	street objects in Texas	14182368/557918	64934
CA	street objects in California	13820481/559349	70584
POIs	POIs worldwide	1157570/1157570	585626

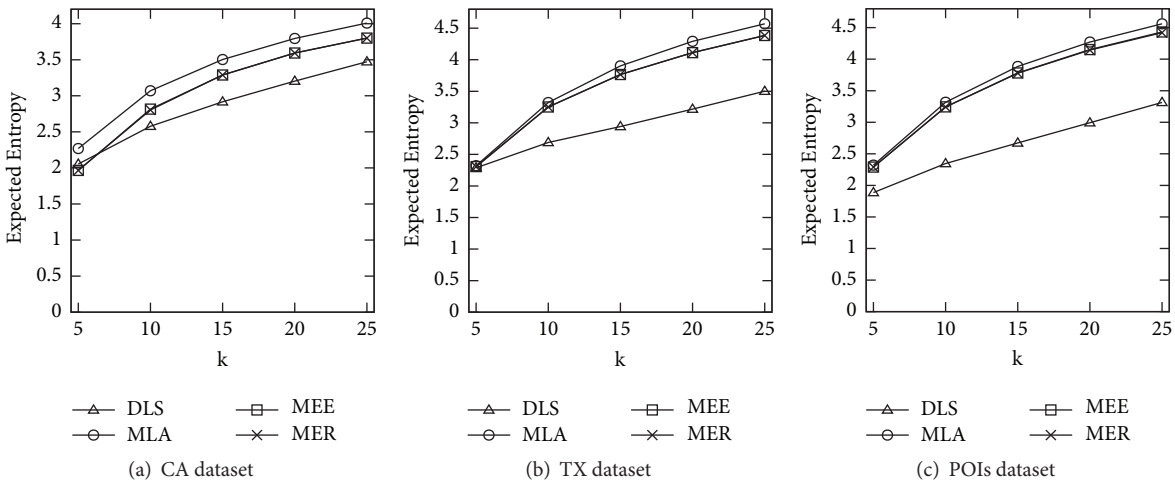


Figure 3: Expected entropy versus parameter k.

a PC-machine with i7-7700 CPU, 8GB memory, and 1TB 7200rpm Hard Disk.

Query Generation. For each prior distribution obtained for a region, we generate 1000 queries, which follows the prior distribution, to test *DLS*. The internal loop times is set to 50 as in [3]. For MLA, *MEE*, and *MER* we directly evaluate the privacy measures using the mechanism obtained for each prior distribution.

Privacy Measures. We employ three privacy measures to evaluate content privacy achieved by MLA and its competitors. These privacy measures are (1) expected entropy; (2) dp-coefficient; and (3) effective k. Expected entropy and dp-coefficient are introduced in Section 3. Effective k measures the number of query contents whose posterior probability is positive, and it measures the uncertainty of the reports in a mechanism.

Parameters. We test the effects of two parameters on the privacy measures we employ. These parameters include the number of query content in a report (denoted k) and the number of query contents in the global set C (denoted N). In the following evaluation k is set to 5, 10, 15, 20, and 25 and its default value is 10. Parameter N is set to 50, 60, 70, 80, 90, and 100 and its default value is 80.

6.2. Evaluation Results. Figures 3 and 4 depict the expected entropy achieved by MLA and its competitors. We first study the effects of parameter k on the expected entropy in Figure 3. Here the total number of query contents in C is set to 80 and k is increased from 5 to 25. As shown in Figure 3, our proposed MLA achieves the best expected entropy in the real-life datasets of *TX*, *CA*, and *POIs*. This is consistent with the fact that MLA achieves the optimal expected entropy. The achieved expected entropy of MLA and its competitors grows with parameter k, since a larger k improves the uncertainty of reports in a mechanism. In the more skewed dataset, i.e., *CA*, MLA outperforms *MEE* and *MER* in larger degree than that of the case in datasets of *TX* and *POIs*. The reason is that MLA splits larger prior probability of query contents into a larger number of reports; thus it is more suitable to deal with skewed prior distribution of query contents. On the other hand, in *MEE*, *MER* and *DLS* (approximately) keep the ratio of posterior probability for two query contents the same as that of their prior probability. Compared to the datasets of *TX* and *POIs*, the expected entropy achieved in *CA* is smaller correspondingly, since more skewed distribution of query content prior probability decreases the optimal expected entropy.

Figure 4 presents the achieved expected entropy when parameter N grows from 50 to 100 while k is fixed at 10. MLA again outperforms its competitors in terms of expected entropy. When N grows, the expected entropy of MLA, *MEE*, and *MER* slightly increases while *DLS* gets decreasing expected entropy. The reason is that increased N brings a relief to the skewness of prior distribution of query contents, so MLA, *MEE*, and *MER* achieve better expected entropy. However, due to the process of *DLS*, querying top frequent contents will make fewer query contents in reports eliminated. When an increasing N relieves the effects of top frequent contents, more query contents in reports of *DLS* get eliminated. Consistent with what is shown in Figure 3, a larger

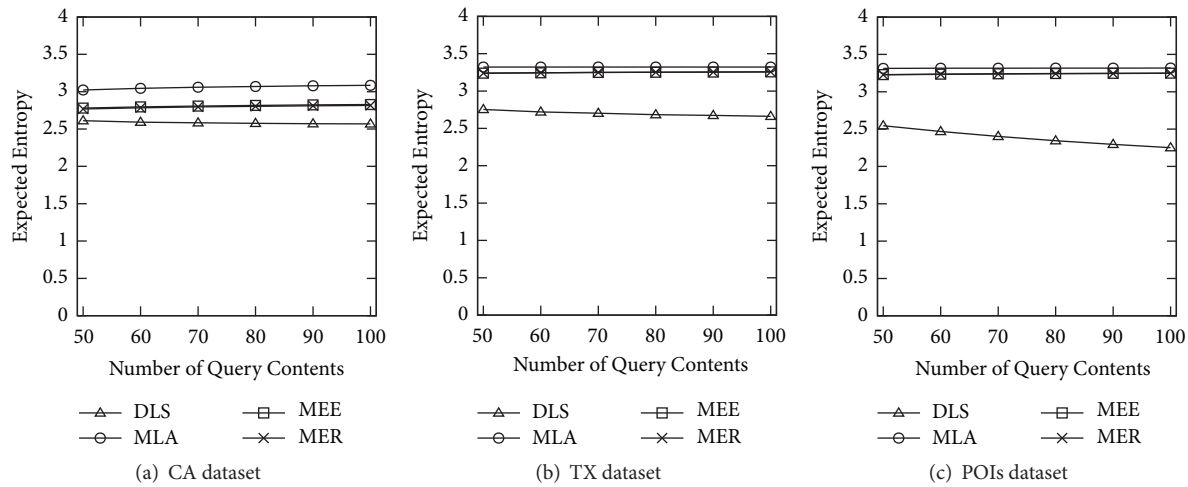


Figure 4: Expected entropy versus parameter N.

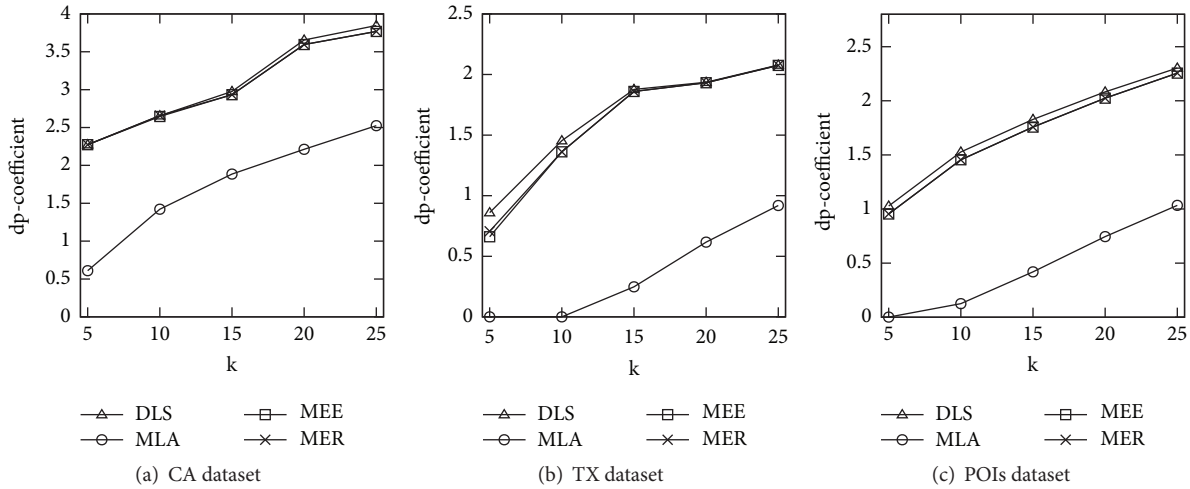


Figure 5: dp-coefficient versus parameter k.

improvement is obtained in *CA* when we compare MLA with *MEE* and *MER*. Meanwhile, better expected entropy is achieved in more uniform datasets of *TX* and *POIs* compared to *CA*.

Next we investigate the dp-coefficient of MLA and its competitors. The privacy measure of dp-coefficient depicts the uncertainty of reports in a mechanism. A smaller dp-coefficient means that it is more difficult for the adversary to eliminate a query content from any report. The effects of parameter k on dp-coefficient is studied in Figure 5. We fix parameter N at 80 and increase k from 5 to 25. In all the datasets, MLA achieves significantly better dp-coefficient compared to *MEE*, *MER*, and *DLS*. When k increases, the dp-coefficient of all the algorithms grows, since more query contents are packed into the same report. In more skewed dataset, *CA*, a larger dp-coefficient is obtained. The skewness increases the difference of prior probability for query contents in the same report. In datasets *TX* and *POIs*, very small dp-coefficient is achieved when k is set to 5 to 10. For other cases of k, the dp-coefficient is almost always smaller than 1, and this means very good uncertainty among query contents in any reports. On the other hand, *MEE*, *MER*, and *DLS* suffer a larger dp-coefficient around 4 and 2 in different datasets, respectively.

The effects of parameter N on dp-coefficient are investigated in Figure 6. When we fix k at 10 and increase N from 50 to 100, *MEE*, *MER*, and *DLS* produce nearly constant dp-coefficient. The dp-coefficient of *MEE*, *MER*, and *DLS* in *CA* is larger than 2.5, while a dp-coefficient around 1.5 is obtained for datasets of *TX* and *POIs*. In contrast, the dp-coefficient of MLA decreases when N grows, since larger N brings relief to the skewness of prior distribution. In *CA*, MLA achieves dp-coefficient smaller than 1.5. For more uniform datasets of *TX* and *POIs*, MLA produces very small dp-coefficients. It obtains ideal dp-coefficient with 0 for *TX* dataset, and the dp-coefficient for *POIs* is also very close to 0. This brings significant difficulty to the adversary to infer the actual query content from any report

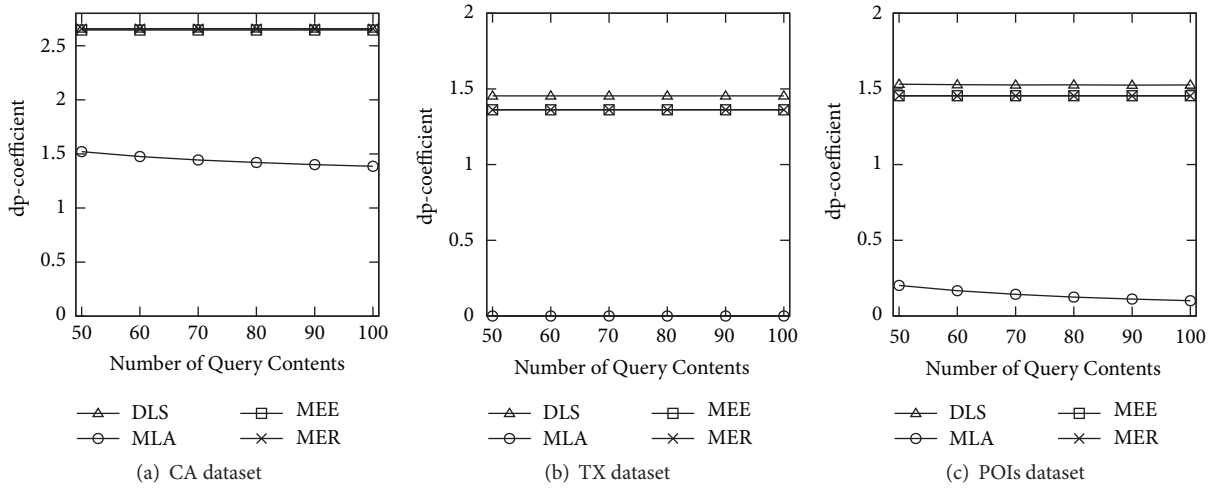


Figure 6: dp-coefficient versus parameter N.

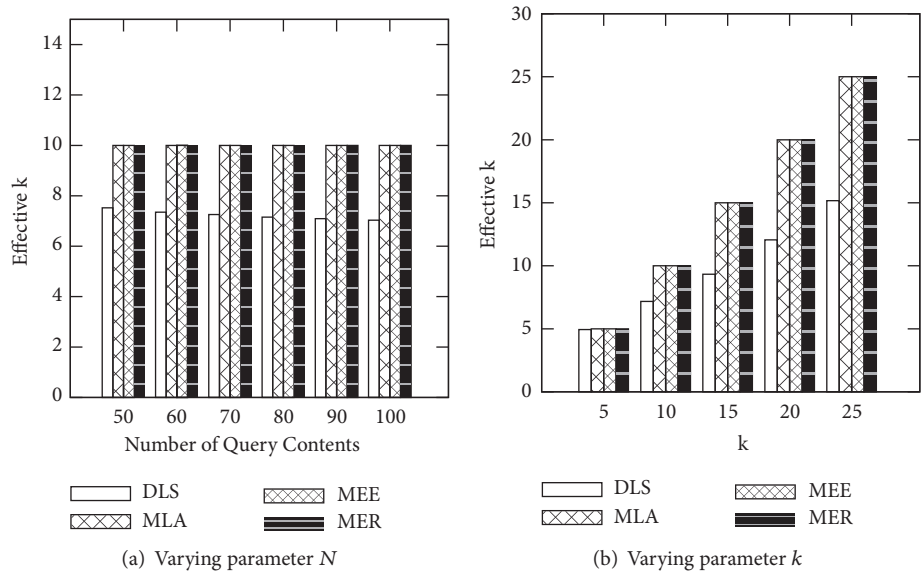


Figure 7: Effective k versus parameters N and k.

of MLA. Generally speaking, MLA achieves much better dp-coefficient compared to *MEE*, *MER*, and *DLS*, and it is able to produce dp-coefficient close to 0 for more uniform datasets.

Finally we test effective k of MLA and its competitors in Figure 7. Given the value of k and N, the same effective k is obtained for different datasets, so we report the effects of parameter N and parameter k on effective k in Figures 7(a) and 7(b), respectively (not for each dataset individually). As shown in Figure 7, MLA, *MEE*, and *MER* achieve the optimal effective k with the value of k. In contrast, *DLS* provides smaller effective k than the value of k. This illustrates the effectiveness of MLA, *MEE*, and *MER* with regard to the disability of eliminating any query contents from each report. We argue that an effective k-anonymity mechanism should provide effective k with the value of k.

In summary, MLA achieves the best privacy measures of expected entropy, dp-coefficient, and effective k simultaneously, which is consistent with our theoretical analysis in Section 5.

7. Related Work

Privacy issues are attracting more and more attention in people's daily life, and studies for protecting privacy in various fields have been proposed, for instance, [11–13] for social network data, [14] for cloud storage, [15–17] for mobile crowd sensing systems, [18, 19] for wireless sensor networks, [1, 20] for sensory data and devices, [21, 22] for cyberphysical systems, and [23] for IoT applications.

Location privacy and content privacy are recognized in location-based services. Solutions to preserving location

privacy and content privacy in location-based services mainly focus on cloaking technique such as [5]. Cloaking technique employs a third-party server to execute spatial generalization algorithms so that the querier is hidden among at least $k-1$ users. However, the third-party server unfortunately possibly becomes the single point of failure for privacy or a performance bottleneck of query processing. To this end, a number of client based solutions [2, 3, 6, 8] are proposed recently. Reference [3] works on the problem of generating proper dummies for locations in reported queries to CPS servers for hiding the user's actual locations. In [3], $2k$ locations with similar probability with the user's location are chosen as dummy candidates, and $k-1$ of them are randomly selected as final dummies. This approach obtains good entropy for the k locations in the reported query. Although this solution includes random nature, the posterior probability of the k reported location is still different due to the process of dummy selection, and the privacy guaranteed is not clear. Reference [6] employs cache to avoid submitting queries to CPS servers as much as possible and thus prevents the leakage of user's location. Reference [2] proposes a mechanism for protecting content privacy in a continuous manner. A set of k query contents are generated for a traveling path, and the user submits the same queries along the path to avoid privacy breach. This fits to continuous querying; however there is no privacy guarantee since it simply chooses query contents with probability larger than a given threshold as candidates. In summary, server-based k-anonymity suffers single point of failure and existing client based solutions do not provide provable privacy guarantee based on the k-location/query contents reported to CPS server. Reference [24] studies improving geoindistinguishability with multiple criteria for better location privacy; however this approach could not be adopted for content privacy due to utility concern. Reference [25] studies protecting privacy for smartphone usage, and this is parallel to our work. Recommendation [26] in location-based system is getting more and more attention, and a location privacy preservation method is proposed for review publication in location-based systems in [27]. The notion of k-anonymity is also developed in statistical databases in [28, 29].

Differential privacy is first introduced in statistic databases [30]. The intuitive idea of differential privacy is that a single change of the input should not modify the output significantly. By this guarantee the adversary cannot recognize the input among all possible inputs similar to the real one. Due to the simple and clean nature of differential privacy, it has been adopted widely, such as machine learning [31], statistic database [32–37], data mining [38], graph [39], data analytic [40], and crowdsourcing [15]. Recent research starts combining correlation [41] and personality [42] nature to original differential privacy. Our work is parallel to the large body of differential privacy research. We combine differential privacy and k-anonymity to provide guaranteed privacy in interactive cyberphysical systems. Differential privacy has been adopted in the literature of privacy protection in location-based services, and [43] ensures that an adversary will not get significant information about a user's location after a query is reported. This is achieved by making the ratio of two nearby locations' posterior probability similar to that of their prior probability. Mechanisms following or adopting similar privacy guarantee are presented to optimize privacy or utility [44]. Besides k-anonymity and differential privacy, a number of works customize semantic privacy metrics such as [45–47] in social networks. This paper also defines privacy measures based on entropy and differential privacy.

8. Conclusion

This paper investigates preserving content privacy in interactive cyberphysical systems through k-anonymity based mechanisms. We present two privacy metrics denoted expected entropy and dp-coefficient, which are based on entropy and differential privacy, respectively, and formulate the problem of achieving the optimal k-anonymity for content privacy in interactive cyberphysical systems based on these privacy metrics. An algorithm MLA consisting of two phases, namely, segment alignment and mechanism initiation, is proposed to establish mechanisms for achieving the optimal k-anonymity. Theoretical analysis illustrates the attractive property that MLA achieves the optimal expected entropy and the optimal dp-coefficient simultaneously. We conduct evaluation based on three real-life datasets, and three privacy metrics, namely, expected entropy, dp-coefficient, and effective k, which depict uncertainty of reports in mechanisms, are tested. Evaluation result demonstrates that MLA outperforms its competitors including recent client based solutions over all the employed privacy metrics, and these results are consistent with the fact that MLA achieves the optimal k-anonymity for content privacy in interactive cyberphysical systems.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by Project (Nos. 61602129, 61772157, and U1509216) supported by the National Natural Science Foundation of China; Heilongjiang Postdoctoral Science Foundation Funded Project (Grant No. LBH-Z14118); Sichuan Science and Technology Foundation funded Project (Grant No. 2017JZ0031).

References

[1] Y Liang I, Zhipeng Cai, Qilong Han, and Yingshu Li, "Location privacy leakage through sensory data," *Security and Communication Networks*, vol. 2017, Article ID 7576307, 12 pages, 2017.

[2] A. Pingley, N. Zhang, X. Fu, H.-A. Choi, S. Subramaniam, and W. Zhao, "Protection of query privacy for continuous location based services," in *Proceedings of the IEEE INFOCOM 2011*, pp. 1710–1718, Shanghai, China, April 2011.

[3] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proceedings of the 33rd IEEE Conference on Computer Communications, IEEE INFOCOM 2014*, pp. 754–762, Toronto, Canada, May 2014.

[4] L. Sweeney, "*k*-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[5] Y. Wang, D. Xu, and F. Li, "Providing location-aware location privacy protection for mobile location-based services," *Tsinghua Science and Technology*, vol. 21, no. 3, pp. 243–259, 2016.

[6] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing privacy through caching in location-based services," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications (IEEE INFOCOM '15)*, pp. 1017–1025, Kowloon, Hong Kong, May 2015.

[7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications and Signal Processing, John Wiley & Sons, New York, NY, USA, 2006.

[8] J. Wang, Y. Li, D. Yang, H. Gao, G. Luo, and J. Li, "Achieving effective k-anonymity for Query privacy in location-based services," *IEEE Access*, vol. 5, pp. 24580–24592, 2017.

[9] Openstreetmap, http://cs.sjtu.edu.cn/~yaobin/sas/#dataset.

[10] Pois dataset, http://www.ntu.edu.sg/home/gaocong/datacode.htm.

[11] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 665–673, 2018.

[12] 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. 577–590, 2018.

[13] X. Zheng, G. Luo, and Z. Cai, "A fair mechanism for private data publication in online social networks," *IEEE Transactions on Network Science and Engineering*, 2018, In Press.

[14] C. Laicheng, L. Yufei, D. Xiaoye, and G. Xian, "User privacy-preserving cloud storage scheme on cp-abe," *Journal of Tsinghua University (Science and Technology)*, vol. 58, no. 2, p. 150, 2018.

[15] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong, and G. Wu, "An incentive mechanism with privacy protection in mobile crowdsourcing systems," *Computer Networks*, vol. 102, pp. 157–171, June 2016.

[16] M. Han, J. Li, Z. Cai, and H. Qilong, "Privacy reserved influence maximization in gps-enabled cyber-physical and online social networks," in *Proceedings of the 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud)*, pp. 284–292, Atlanta, Ga, USA, 2016.

[17] M. Han, Y. Li, J. Li, L. Li, and Q. Han, "Maximizing influence in sensed heterogenous social network with privacy preservation," *International Journal of Sensor Networks*, vol. 1, no. 1, article 1, 2017.

[18] H. Huang, T. Gong, P. Chen, R. Malekian, and T. Chen, "Secure two-party distance computation protocol based on privacy homomorphism and scalar product in wireless sensor networks," *Tsinghua Science and Technology*, vol. 21, no. 4, pp. 385–396, 2016.

[19] M. Dapeng, W. Chenye, Y. Wu, W. Wei, X. Shichang, and J. Xiaopeng, "Energy-efficient cluster-based privacy data aggregation for wireless sensor networks," *Journal of Tsinghua University (Science and Technology)*, vol. 57, no. 2, p. 213, 2017.

[20] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[21] X. Zheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Follow but no track: privacy preserved profile publishing in cyber-physical social systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1868–1878, 2017.

[22] 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*, 2018, In Press.

[23] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart iot systems: a consideration from privacy perspective," *IEEE Communications Magazine*, 2018, In press.

[24] S. Oya, C. Troncoso, and F. Pérez-González, "Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1959–1972, Dallas, Tex, USA, October 2017.

[25] L. Zhang, Z. Cai, and X. Wang, "FakeMask: a novel privacy preserving approach for smartphones," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 335–348, 2016.

[26] Z. Zhou, Z. Cheng, L.-J. Zhang, W. Gaaloul, and K. Ning, "Scientific workflow clustering and recommendation leveraging layer hierarchical analysis," *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 169–183, 2018.

[27] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *Proceedings of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, Ga, USA, May 2017.

[28] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, article 3, Article ID 1217302, 2007.

[29] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: privacy beyond k-anonymity and *l*-diversity," in *Proceedings of the 23rd International Conference on Data Engineering*, pp. 106–115, IEEE, Istanbul, Turkey, April 2007.

[30] C. Dwork, "Differential privacy," in *ICALP 2006: Automata, Languages and Programming*, vol. 4052 of *Lecture Notes in Computer Science*, pp. 1–12, Springer, Berlin, Germany, 2006.

[31] K. Boyd, E. Lantz, and D. Page, "Differential privacy for classifier evaluation," in *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security, AISec 2015 - co-located with CCS 2015*, pp. 15–23, Denver, Colo, USA, 2015.

[32] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proceedings of the 2011 ACM SIGMOD and 30th PODS 2011 Conference on Management of Data (SIGMOD)*, pp. 193–204, Athens, Greece, June 2011.

[33] X. He, A. Machanavajjhala, and B. Ding, "Blowfish privacy: Tuning privacy-utility trade-offs using policies," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, (SIGMOD '14)*, pp. 1447–1458, Snowbird, Utah, USA, June 2014.

[34] D. Kifer and A. Machanavajjhala, "Pufferfish: a framework for mathematical privacy definitions," *ACM Transactions on Database Systems (TODS)*, vol. 39, no. 1, article 3, 2014.

[35] S. Haney, A. Machanavajjhala, and B. Ding, "Design of policy-aware differentially private algorithms," *Journal Proceedings of the VLDB Endowment*, vol. 9, no. 4, pp. 264–275, 2015.

[36] S. Song, Y. Wang, and K. Chaudhuri, "Pufferfish privacy mechanisms for correlated data," in *Proceedings of the the 2017 ACM International Conference*, pp. 1291–1306, Chicago, Ill, USA, May 2017.

[37] S. Haney, A. Machanavajjhala, J. M. Abowd, M. Graham, M. Kutzbach, and L. Vilhuber, "Utility cost of formal privacy for releasing national employer-employee statistics," in *Proceedings of the the 2017 ACM International Conference*, pp. 1339–1354, Chicago, Ill, USA, May 2017.

[38] A. Friedman and A. Schuster, "Data mining with differential privacy," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-2010*, pp. 493–502, Washington, Wash, USA, July 2010.

[39] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith, "Analyzing Graphs with Node Differential Privacy," in *TCC 2013: Theory of Cryptography*, vol. 7785 of *Lecture Notes in Computer Science*, pp. 457–476, Springer, Berlin, Germany, 2013.

[40] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton, "Bolt-on differential privacy for scalable stochastic gradient descent-based analytics," in *Proceedings of the the 2017 ACM International Conference*, pp. 1307–1322, Chicago, Ill, USA, May 2017.

[41] C. Liu, S. Chakraborty, and P. Mittal, "Dependence makes you vulnerable: differential privacy under dependent tuples," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, Calif, USA, 2016.

[42] Z. Jorgensen, T. Yu, and G. Cormode, "Conservative or liberal? Personalized differential privacy," in *Proceedings of the 2015 31st IEEE International Conference on Data Engineering, ICDE 2015*, pp. 1023–1034, Seoul, South Korea, April 2015.

[43] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: differential privacy for location-based systems," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*, pp. 901–914, Berlin, Germany, November 2013.

[44] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 21st ACM Conference on Computer and Communications Security, CCS 2014*, pp. 251–262, Scottsdale, Ariz, USA, November 2014.

[45] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, "Cost-efficient strategies for restraining rumor spreading in mobile social networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2789–2800, 2017.

[46] Z. He, Z. Cai, and X. Wang, "Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks," in *Proceedings of the 35th IEEE International Conference on Distributed Computing Systems (ICDCS '15)*, pp. 205–214, Columbus, Ohio, USA, July 2015.

[47] Z. He, Z. Cai, Y. Sun, Y. Li, and X. Cheng, "Customized privacy preserving for inherent data and latent data," *Personal and Ubiquitous Computing*, vol. 21, no. 1, pp. 43–54, 2017.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 3980371, 10 pages
https://doi.org/10.1155/2018/3980371

Research Article

User Presence Inference via Encrypted Traffic of Wireless Camera in Smart Homes

Xiaoyu Ji, Yushi Cheng, Wenyuan Xu, and Xinyan Zhou

Ubiquitous System Security Lab (USSLAB), Zhejiang University, Hangzhou 310027, China

Correspondence should be addressed to Wenyuan Xu; wyxu@zju.edu.cn

Received 15 June 2018; Accepted 6 September 2018; Published 25 September 2018

Guest Editor: Wei Li

Wireless cameras are widely deployed in smart homes for security guarding, baby monitoring, fall detection, and so on. Those security cameras, which are supposed to protect users, however, may in turn leak a user's personal privacy. In this paper, we reveal that attackers are able to infer whether users are at home or not, that is, the user presence, by eavesdropping the traffic of wireless cameras from distance. We propose `HomeSpy`, a system that infers user presence by inspecting the intrinsic pattern of the wireless camera traffic. To infer the user presence, `HomeSpy` first eavesdrops the wireless traffic around the target house and detects the existence of wireless cameras with a Long Short-Term Memory (LSTM) network. Then, `HomeSpy` infers the user presence using the bitrate variation of the wireless camera traffic based on a cumulative sum control chart (CUSUM) algorithm. We implement `HomeSpy` on the Android platform and validate it on 20 cameras. The evaluation results show that `HomeSpy` can achieve a successful attack rate of 97.2%.

1. Introduction

Home security cameras are widely deployed in smart homes to provide protection services ranging from home security through baby monitoring to fall detection. Wi-Fi wireless cameras, that is, Internet protocol cameras equipped with Wi-Fi modules, are booming among home security camera market due to their flexibility and usability. According to Technavio [1], the global wireless video surveillance market will continue to grow at a rate of 21.35% during the period 2014–2019. However, despite its popularity and convenience, we discover that wireless cameras, which are designed to provide protection services, are likely to leak personal privacy and become a point of attacking.

Already, an increasing attention has been paid to the privacy issues caused by wireless cameras [2–4]. Much work has been proposed to safeguard personal privacy against cameras. This work [5] introduces an "invisible light beacon" implemented on the eye-wear to prevent unauthorized video-taping, by which the privacy preferences of photographed users are communicated to photographing cameras. These authors [6–9] focus on the privacy concerns caused by "first-person" wearable cameras. They propose methods to identify and prevent the sharing of sensitive images captured by wearable cameras. Birnbach et al. [10] detect drones carrying out privacy invasion attacks with on-board cameras. This work analyzes the RSSI (received signal strength indicator) of the wireless traffic from the camera on the drone to detect its approach.

Different from previous work, our paper focuses on the wireless security cameras and reveals that wireless cameras may leak user privacy, that is, the user presence. We propose `HomeSpy`, a system that is able to infer whether users are at home or not, by eavesdropping the wireless camera traffic. As shown in Figure 1, an adversary tries to figure out whether there are any users inside a target house. She runs `HomeSpy` on her smartphone outside the house to infer the user presence. `HomeSpy` overhears and analyzes the wireless traffic, even it is encrypted, and informs the adversary that the owner is likely not at home. With this information, the adversary can conduct further malfeasance, that is, wade the house physically.

Inferring user presence via wireless camera traffic is promising yet challenging. Since an adversary has no access to the house as well as its Wi-Fi network, `HomeSpy` shall work without joining the network. In addition, there might

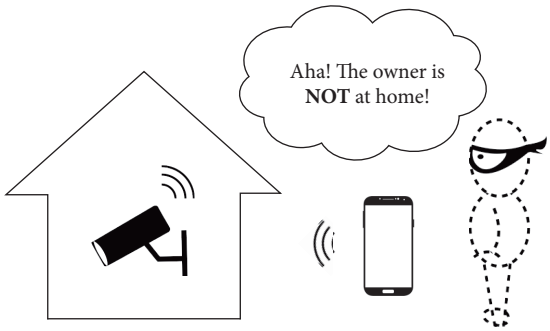


Figure 1: HomeSpy is able to infer the user presence by eavesdropping the wireless camera traffic.

be various devices inside the house that generate wireless traffic; HomeSpy shall figure out whether there is a wireless camera and which packets belong to the wireless camera. Furthermore, camera traffic is encrypted and thus traditional image/audio processing techniques are invalid in this scenario. HomeSpy shall exploit a new angle to reveal the information ensconced in the camera traffic.

Our key insight is that compression and fragmentation over video/audio streams have fundamental impact on the traffic of wireless cameras, which appears as a sequence of full-size packets with small-size packets in between. Such a traffic pattern shall make the wireless camera distinguishable from other network applications. In addition, the number and size of video/audio frames of wireless camera traffic are not fixed and depend on the video/audio content. If a user is within the filming range of the wireless camera, the frame number and size are likely to change with the human intervention (i.e., motion and sound). By exploring the human intervention in the wireless camera traffic, HomeSpy shall be able to infer whether the user is at home or not, that is, the user presence.

To overcome all the aforementioned challenges, HomeSpy eavesdrops the wireless traffic near the target house with a smartphone and detects the existence of wireless cameras inside the house with a LSTM network. Then, HomeSpy inspects the bitrate variation of the camera traffic and infers the user presence with a CUSUM algorithm. In summary, our contribution includes the following.

(i) We reveal that wireless surveillance cameras in smart homes are potential sources of privacy leakage, that is, the user presence status, which can result in serious security issues.

(ii) We propose HomeSpy, a system that is able to analyze whether owners are at home or not, by eavesdropping the network traffic of wireless security cameras.

(iii) We implement HomeSpy on the Android platform and validate it on 20 popular wireless security cameras. The results demonstrate that HomeSpy can successfully attack with a probability of 97.2%.

The rest of the paper proceeds as follows. First, we outline the background and insights of HomeSpy in Section 2. Then, we present the threat model of HomeSpy in Section 3. In Section 4, we elaborate the design of HomeSpy in detail, and in Section 5, we present the implementation details and experimental results. Discussion and related work are provided in Sections 6 and 7, respectively, with Section 8 giving concluding remarks.

2. Background

In this section, we first outline the basics of the wireless camera and, then, its intrinsic traffic patterns exploited by HomeSpy.

2.1. Basics of Wireless Camera. Wireless security cameras provide real-time video surveillance through a wireless network operating at license-free frequencies (e.g., 2.4 GHz). Instead of saving records locally, wireless cameras process the video/audio streams and then upload them through an access point (AP) of a wireless local area network (WLAN) to a remote cloud server. Modern wireless cameras, therefore, allow users to achieve remote monitoring by accessing the cloud server.

The hardware modules of a wireless camera are shown in Figure 2. Wireless cameras monitor target environments, for example, homes and offices, with build-in CMOS/CCD sensors, which capture and digitize the image scenes to generate raw video and audio frames. The raw multimedia frames are then fed into an SOC (system on chip) before being transmitted to a remote client or server by the Wi-Fi chip. A multimedia SOC chip usually contains a MCU and three additional submodules: (1) image signal processing (ISP) submodule that performs functions such as noise filtering; (2) codec submodule that compresses video/audio frames to decrease frame sizes; (3) networking protocol submodule that transmits multimedia stream via streaming protocols like RTSP (real-time streaming protocol).

2.2. Wireless Camera Traffic Pattern. As mentioned above, the codec submodule compresses video/audio frames to decrease frame sizes. Various brands of cameras may have slightly different implementation, but most of them employ popular compression standards, for example, H.264 [11] for video and AAC [12] for audio. H.264 standard utilizes the redundancy between consecutive frames and outputs a series of Intra coded frames (I-frame), Predicted frames (P-frame), and Bi-directional predicted frames (B-frame), for example, a sequence of $\{I, P, B, P, B, P, B, I, \ldots\}$ [13], as shown in Figure 3. I frames are coded without reference and can be decoded by themselves. P frames are coded based on the prior frames, while B frames are coded based on both the prior and the subsequent frames. The number of P and B frames between two consecutive I frames is not fixed and depends on the video content. Since I frames do not utilize inter-frame redundancy, they typically are a few times larger than the other two frame types, and P frames are larger than B frames. An audio stream is compressed similarly but with a much lower bandwidth. Audio signals are sampled continuously, and encoding protocols (e.g., AAC) are used on the audio samples to remove redundancy.

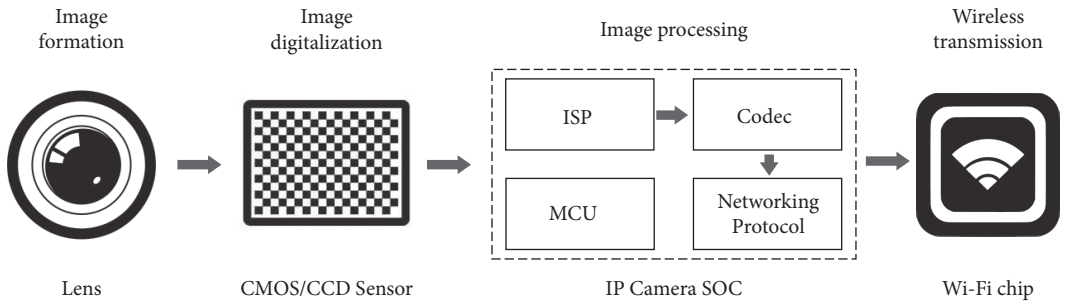


Figure 2: Hardware modules inside a wireless camera.

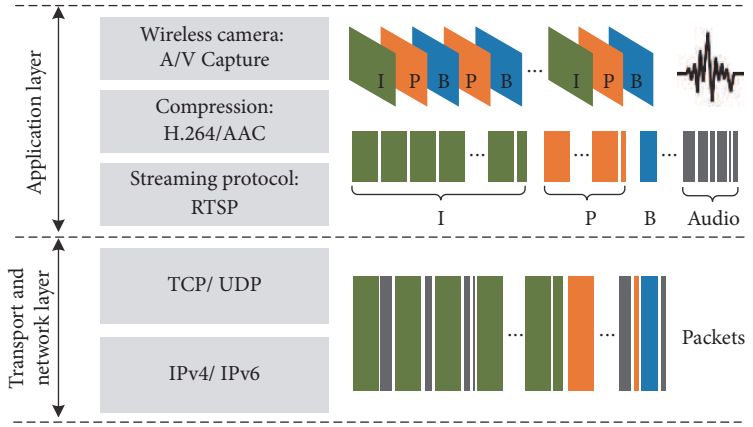


Figure 3: Compression and fragmentation of wireless camera stream.

After being compressed, the video and audio stream frames go through the application layer to be encapsulated with networking protocols, such as RTSP, before being transmitted in the data link layer. Since the network interface in the link layer can only transmit a packet of a size smaller than its maximum transmission unit (MTU), when a frame is larger than MTU, it has to be fragmented. For video streams, since I frames are typically the largest among the three types, one I frame is divided into the largest number of packets. Denote the number n, and then the first $n - 1$ packets are *full-size packets* (i.e., MTU) and the last one may be smaller than the full size, which we call *small-size packet*. B frames are typically the smallest, and they may fit in one packet or a few packets. In comparison, the frames of audio streams are smaller than videos and thus, a less number of full-size packets are formed. In general, video and audio frames are divided into both full-size packets and small-size packets as follows:

(i) I frames are divided into a set of full-size packets followed by a small-size packet.

(ii) P frames are divided into a smaller set of full-size packets followed by a small-size packet.

(iii) Most B frames and audio frames contribute to small-size packets.

As such, compression and fragmentation over video/audio streams form a unique traffic pattern, which can be exploited to differentiate wireless cameras from other network applications.

2.3. Human Impact on Wireless Camera. For the sake of privacy protection, the camera traffic is encrypted in the SOC chip. Therefore, an adversary cannot obtain the video/audio content from the traffic directly. However, after conducting an in-depth study on the working principle of wireless cameras, we find that even if encrypted, the camera traffic can still reveal privacy information.

As mentioned above, the number and size of video frames are not fixed and depend on the video content. When the video content changes, the frame number and size increase to capture the discrepancy of the content. Therefore, if a user is within the filming range of the wireless camera, the number and size of the frames are likely to change with the human motion. Similarly, the voice from the user also increases the audio traffic. Thus, both motions and voices from a user can increase the frame number and size of the wireless camera, resulting in a larger amount of traffic and thus a higher bitrate. This finding sheds light upon the insights of `HomeSpy`. As humans may impact the traffic of the inside wireless camera, the bitrate of the wireless camera may in turn reveal whether there is any person inside the house.

3. Threat Model

In this section, we present the threat model of `HomeSpy`. Since the adversary's goal is to infer user presence from distance, we consider the following attack scenario: *in a residential area, an adversary attempts to commit crimes, such as burglary. To*

reduce the chance of getting caught, the adversary tends to choose houses without the presence of owners. To achieve it, the adversary utilizes HomeSpy *to infer the user presence status*. In such a scenario, we assume the adversary has following abilities.

Vicinity to Target House. We assume that the adversary may target any houses of their choices and can be in the target house's vicinity and draw no attention.

No Prior Information of Wireless Camera. Although the adversary may obtain some prior knowledge of the target house, to increase the attack difficulty, we assume they have no information whether there is any wireless camera inside the house.

No Access to Wi-Fi Network. We assume the adversary has no password for the Wi-Fi network in the target house. Thus, they have no access to the Wi-Fi network and cannot collect the wireless traffic by joining the network.

No Specific User Interaction. The adversary is unable to ask users to perform any operations. Thus, they cannot rely on any specific user operations, for example, making a phone call or running a specific application, but natural user behaviors that are introduced frequently at home.

4. Attack Design

Based on the threat model mentioned above, we design HomeSpy to enable the adversary to infer user presence from distance. In this section, we elaborate the attack design in detail.

4.1. Workflow of HomeSpy. The workflow of HomeSpy is composed of four steps: traffic eavesdropping, flow construction, camera detection, and bitrate analysis. First, HomeSpy sets the Wi-Fi card on the smartphone into the monitor mode to capture network traffic over wireless channels near the target house. Then, HomeSpy filters non-data and downlink packets and constructs the remaining packets into flows based on their MAC addresses. In the camera detection step, HomeSpy feeds the packet length sequence of each flow into a LSTM network and detects the existence of wireless cameras. Finally, HomeSpy examines the bitrate variation of the detected camera flow with a CUSUM algorithm to infer the user presence.

4.2. Traffic Eavesdropping. A straightforward method to overhear the network traffic is to join the network; however, an adversary usually has no access to the Wi-Fi password; even if they do, most WLANs (wireless local area networks) are encrypted with WPA/WPA2-PSK [14]. These methods exploit per-client, per-session keys, which are derived from the Wi-Fi password and the information exchanged when a client joins the network [15]. As a result, even with the Wi-Fi password, it is difficult for the adversary to capture all four handshake packets to derive the keys.

To eavesdrop the wireless traffic of the target house without joining the network, HomeSpy collects data by setting the smartphone's Wi-Fi card into the monitor mode. Normally, a smartphone sets its Wi-Fi card in the managed mode, in which packets not destined for this smartphone are discarded. However, HomeSpy requires to eavesdrop all the nearby wireless traffic for analysis. To this end, the Wi-Fi card shall be set into the monitor mode such that HomeSpy is able to capture and record all the passing wireless packets.

We implement the monitor mode function on the Android platform based on an open-source project named Nexmon [16]. Nexmon provides a basic API for Wi-Fi driver modification. We use a UDP socket to read packets from the `rawproxy` application, which connects to the Wi-Fi card buffer in Nexmon. Once receiving a packet via the UDP socket, the packet is decoupled and collected for further analysis. In this way, HomeSpy is able to eavesdrop the network traffic on the sly.

4.3. Flow Construction. In the flow construction step, HomeSpy first filters packets that are unlikely to be from camera applications. Since wireless cameras mainly create uploading streams with a destination of an access point, we remove both non-data packets and downlink packets. Non-data packets include ACK, RTS, CTS, and other management packets, and downlink flows are identified by reading the `FC` field in the MAC frame header. HomeSpy only records the `length` and `address` fields of each packet and discards the payload, for the sake of efficiency.

Then, HomeSpy groups the remaining packets into flows according to their MAC addresses. In this step, we put packets with identical source and destination MAC addresses into the same group and regard all packets belonging to the same MAC pair as an individual flow.

4.4. Camera Detection. As revealed in Section 2, compression and fragmentation over video/audio streams have fundamental impact on the wireless camera traffic pattern. The H.264 and AAC standards indicate that when video/audio streams are passed to the network interface cards, a sequence of I, P, B, and audio frames are packaged into a series of full-size packets with small-size packets in between.

HomeSpy detects the wireless camera flow by exploiting this packet length pattern. However, since different cameras have various bitrates and resolutions, and the packet length sequences of different applications can diverse significantly in terms of length even within the same time period, it is difficult to extract robust features manually. We build a Long Short-Term Memory network [17] that excels at dealing with time sequences and can learn discriminative features from samples by itself. The built LSTM network takes the packet length sequence of each flow as the input and outputs whether it is a wireless camera flow. This completely data driven approach, compared to traditional hand crafted feature extraction methods, leads to much simpler-to-use and more reliable outcomes in practice.

Moreover, to lighten the network overhead and achieve real-time attacks, we implement the LSTM network on the smartphone instead of uploading raw data to the cloud and running classification algorithms remotely. Due to the limited resources and computational capability on the mobile

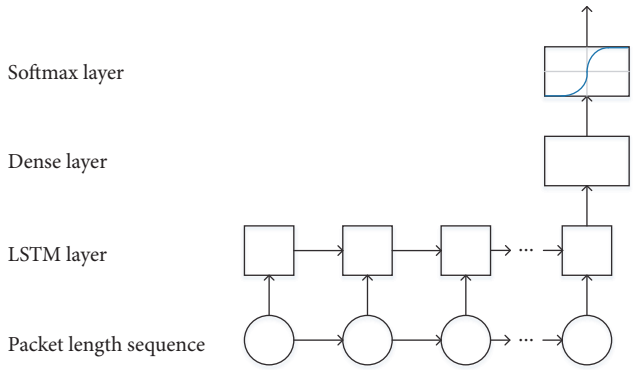


Figure 4: Network architecture of the proposed LSTM model.

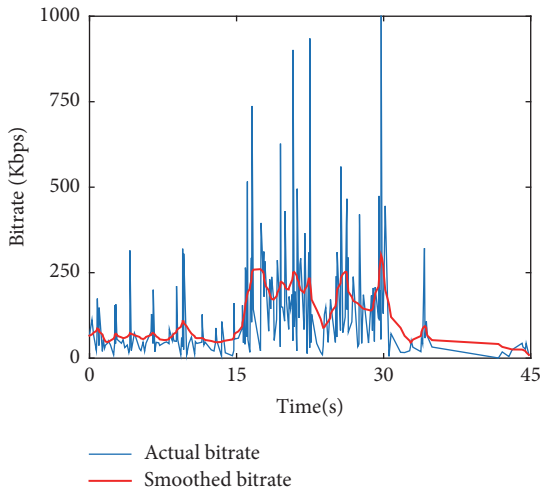


Figure 5: Impact of human beings on the bitrate of the camera flow.

devices, the LSTM network employed on the smartphone shall be lightweight. The architecture of the employed LSTM network is shown in Figure 4, where we only use one LSTM layer for the sake of lightweight. In addition, we elaborately select the parameters of the network in Section 5, that is, the number of neurons in the LSTM layer $m = 16$, the number of neurons in the dense layer $n = 16$, the times of iteration $k = 5$, and the size of batching $b = 64$, to strike the balance of accuracy and efficiency on the smartphone.

Note that this wireless camera detection approach targets nonintensive houses. For intensive residences, for example, apartments, we can further infer in which room the wireless camera is installed by widely studied wireless localization techniques [18–22].

4.5. Bitrate Analysis. As mentioned in Section 2, if a person is within the filming range of the wireless camera, the frame number and size will change with the human intervention, as well as the bitrate. An illustration is shown in Figure 5, in which a human keeps moving during the period from 15 *s* to 30 *s*. As a result, the bitrate of the camera flow also shows a rise and fall during this period with only a one-second lag. With this observation, we employ the bitrate variation of the detected camera flow to infer the user presence status: present or absent.

To detect the changes of bitrate caused by human beings, HomeSpy utilizes the cumulative sum control chart (CUSUM) [23] algorithm to detect the rising edges of the bitrate sequence r. CUSUM is a sequential analysis technique typically used for change monitoring. The CUSUM algorithm for bitrate variation detection is as follows:

$$U_n = \begin{cases} 0, & n = 0 \\ \max\left(0, U_{n-1} + r_n - w_n\right), & n > 0 \end{cases} \quad (1)$$

$$Condition : U_n > \delta, \quad n = 0, 1 \ldots N \quad (2)$$

where U_n is the upper cumulative sum at the time n. w is the likelihood estimation of the bitrate sequence. δ is the threshold for detecting inside human beings. If the value of U_n exceeds δ, HomeSpy outputs that the owners of the house are at home; otherwise, HomeSpy believes they may have gone out.

5. Performance Evaluation

In this section, we evaluate the performance of HomeSpy. We first present the setup of the experiments, then the evaluation metrics, and finally the results of HomeSpy attack.

5.1. Experimental Setup

Experiment Scene. We perform experiments in a house with a wireless camera installed in the bedroom. The adversary is outside the house with a distance of 3 *m* and runs the HomeSpy application on an LG Nexus 5 smartphone in real-time to infer the user presence. Note that the attack distance is set as 3 *m* for illustration but not limited to it. In fact, HomeSpy is effective as long as the attacker can capture the wireless traffic of the home security camera. Therefore, the effective attack distance of HomeSpy roughly equals to the Wi-Fi covering range.

Camera Brands. We select 20 typical cameras of 12 well-known manufacturers on the market from both US and China in our experiments, and their detailed information including brand, model, compression protocol, and default resolution is summarized in Table 1.

5.2. Performance Metrics. We use *precision*, *recall*, and *F1-score* to evaluate the performance of HomeSpy on wireless camera detection and utilize *true positive rate (TPR)*, *true negative rate (TNR)*, and *successful attack rate (SAR)* to evaluate the performance of HomeSpy on user presence inference.

Precision. We define precision as

$$precision = \frac{TP}{FP + TP} \quad (3)$$

where TP and FP represent the true positives and the false positives of wireless camera detection, respectively.

Table 1: Summary of experimental cameras.

No.	Brand	Model	Compression	Default Resolution
1	Dahua	LeChange TC1	H.264, AAC	720p
2	Dahua	LeChange TC5	H.265, AAC	1080p
3	Dahua	LeChange TC6C	H.264, AAC	720p
4	Dahua	LeChange TC7C	H.264, AAC	720p
5	Hikvision	Ezviz C2C	H.264, AAC	720p
6	Hikvision	Ezviz C2mini	H.264, AAC	720p
7	Yi	1	H.264, AAC	720p
8	Yi	2	H.264/MJPEG, AAC	1080p
9	Xiaomi	iSC5	H.264, AAC	1080p
10	Xiaomi	SXJ01ZM	H.264, AAC	1080p
11	360	D600	H.264, Opus	720p
12	360	D606	H.264, Opus	1080p
13	TP-LINK	TL-IPC20-2.8	H.264, -	720p
14	TP-LINK	TL-IPC10A	H.264, AAC	720p
15	Nest	Cam Indoor	H.264, AAC	1080p
16	Amcrest	IP2M-841W	H.264, AAC	1080p
17	D-Link	DCS-820L	H.264/JPEG, AAC	1080p
18	Lenovo	G2 Mini	H.264/JPEG, AAC	720p
19	Haier	HC6700	H.265/JPEG, AAC	720p
20	Yoosee	KP-01	H.264, AAC	720p

Recall. We define recall as

$$recall = \frac{TP}{FN + TP} \tag{4}$$

where FN is the false negatives of wireless camera detection.

F1-Score. We define F1-score as

$$F1 - score = \frac{2 \times (precision * recall)}{precision + recall} \tag{5}$$

to strike the balance of precision and recall.

Successful Attack Rate. We define TPR and TNR as the rate of correctly inferring user presence and absence, respectively. Since HomeSpy attacks the wireless camera to infer the user presence, both true positive and true negative are successful attacks. Hereby, we define

$$SAR = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{6}$$

as the successful attack rate.

5.3. Wireless Camera Detection Performance. In the first set of experiments, we evaluate the wireless camera detection performance of HomeSpy with aforementioned 20 cameras. For each camera, we collect 100 traces (each lasting 10 s) when the camera working inside the house with 7 other network devices nearby (e.g., mobile phones, laptops, and tablets), which also generate wireless traffic. Note that we regard each camera flow as a positive sample and each non-camera flow as a negative sample. Thus, a collected trace contains a positive sample and several negative samples.

We randomly choose 10% of the samples as the training set, and the remaining samples serve as the testing set. To adapt to the source limited environment on smartphones, we evaluate the appropriate size of m (the number of neurons in the LSTM layer), n (the number of neurons in the dense layer), k (the times of iteration), and b (the size of batching). The results are demonstrated in Figures 6–8, from which we have following findings. First, with the increase in m and n, the performance of HomeSpy is slightly enhanced at a cost of a larger network and more calculation. We set m and n both to be 16 to obtain a small but efficient network on smartphones. Second, with the increase in k, the performance of HomeSpy is slightly improved at a cost of a larger computation delay, while a larger batching size b can reduce the delay at a cost of performance decrease. To balance the computation delay and accuracy, we set the $k = 5$ and $b = 64$. With the selected parameters, HomeSpy can achieve an overall wireless camera detection performance of 98.1% precision, 98.3% recall, and 98.2% F1-score across 20 cameras on smartphones.

5.4. User Presence Inference Performance. In the second set of experiments, we evaluate the user presence inference performance of HomeSpy with 3 cameras (Ezviz, Dahua, and Yi). Three volunteers participated in the experiments. To simulate the present status, the volunteers are asked to walk around for 15 seconds inside the house. For the absent status, the house has no human beings inside. For each status, we utilize HomeSpy for inference and repeat this process for 30 times.

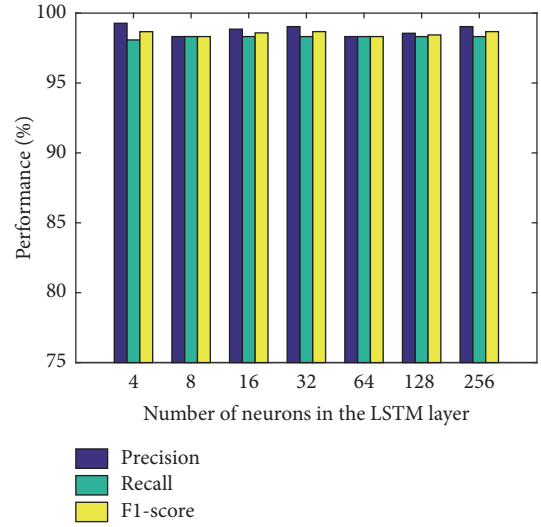


Figure 6: Impact of the number of neurons in the LSTM layer m (with $n = 32$, $k = 10$, and $b = 32$).

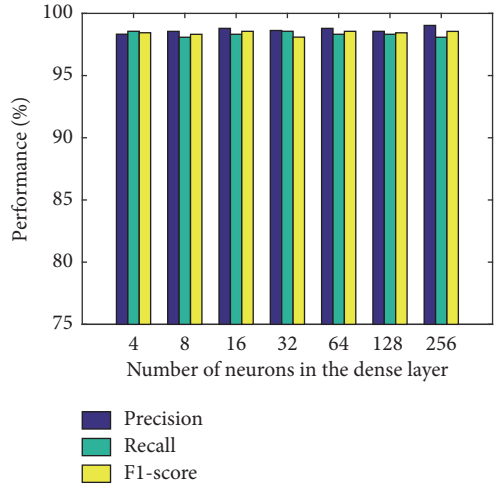


Figure 7: Impact of the number of neurons in the dense layer n (with $m = 16$, $k = 10$, and $b = 32$).

The results in Figure 9 reveal that HomeSpy can successfully attack with an average SAR of 97.2%. False positives mainly come from the fluctuation of bitrate resulting from the dynamic of network environment. Based on the ROC (receiver operating characteristic) curve in Figure 10, we utilize a large threshold and sacrifice a bit of accuracy to achieve a low false positive rate.

Next, we consider several factors that could affect our ability to infer the user presence.

5.4.1. Impact of Motion Range. Since the bitrate variation has resulted from human motions, the motion range may affect the inference performance. A larger motion range can result

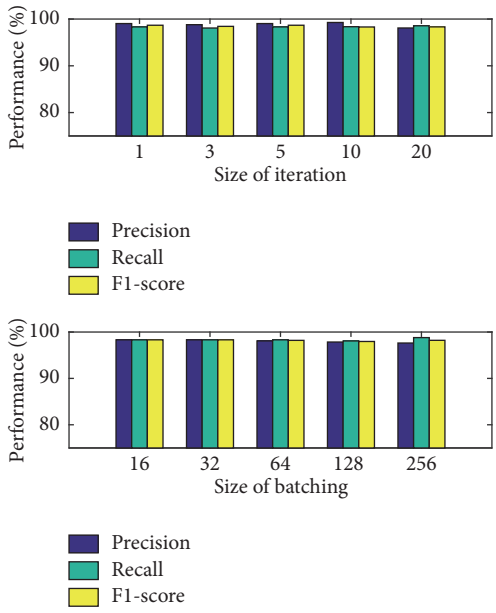


Figure 8: Impact of the iteration times k (with $m = 16$, $n = 16$, and $b = 32$) and the batching size b (with $m = 16$, $n = 16$, and $k = 5$).

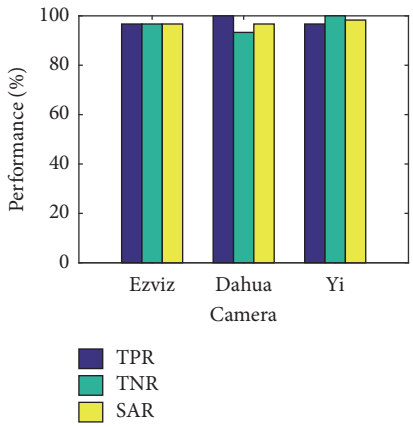


Figure 9: Overall user presence inference performance of HomeSpy.

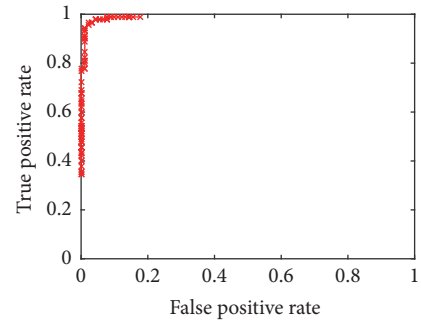


Figure 10: The ROC curve of user presence inference.

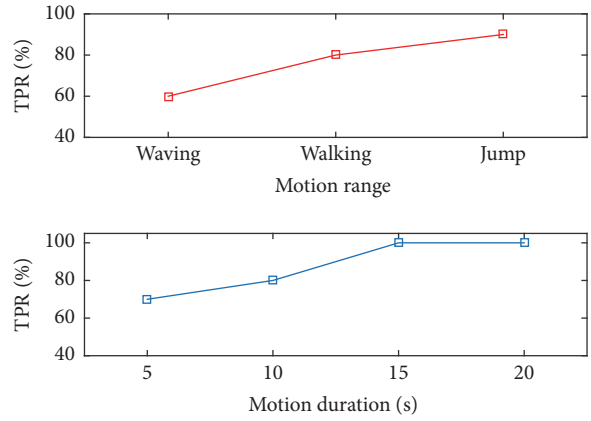


Figure 11: The impact of motion range and motion duration.

in more bitrate variation and, thus, may improve the inference accuracy.

To investigate the impact of motion range, we conduct experiments with three common motions: waving (wave hands), walking (walk around the room), and jump (jump up and down). Apparently, the intensity of motion is strengthened from front to back. The motion duration is set to 10 *s* and 10 attacks are launched for each motion. The results in Figure 11 show that the TPR does increase with the growth of motion range. Even with slight movements such as waving hands, HomeSpy can still successfully attack with a rate of 60%, while jump holds the highest probability of 90%.

5.4.2. Impact of Motion Duration. Another influencing factor of HomeSpy is the motion duration time. Long duration of human intervention helps overcome the transient variation of bitrate caused by the dynamic network environment and, thus, may decrease the false positive rate.

To evaluate the impact of motion duration, we conduct experiments with four typical time durations: 5 *s*, 10 *s*, 15 *s* and 20 *s*. The results are also revealed in Figure 11, which confirm that the longer the time duration is, the better the attack accuracy will be. Even with only 5 *s* of human intervention, HomeSpy is able to successfully attack with a rate of 70%. With the increase in the duration time, the TPR is promoted to 100% for 15 *s* as well as 20 *s*.

5.4.3. Impact of Household Pet. In real life, many families have pet dogs or cats. It is likely that household pets introduce motion or sound as well and, thus, cause the variation of wireless camera traffic. To investigate the impact of pets on the user presence inference, we perform HomeSpy attack with a medium-size household dog. For the present status, both the volunteer and the dog are inside the house, whereas only the dog stays in the room. For each status, we utilize HomeSpy for inference and repeat this process 30 times.

The results in Figure 12 reveal that the pet dog does not impact the true positive rate of HomeSpy (97.8% on average). However, false positives rise due to the impact of household pets, which decrease the true negative rate. Nevertheless, even with the interference from a household pet, HomeSpy can successfully attack with an average SAR of 91.1%. In addition, we argue that as the goal of the attacker is to commit crimes,

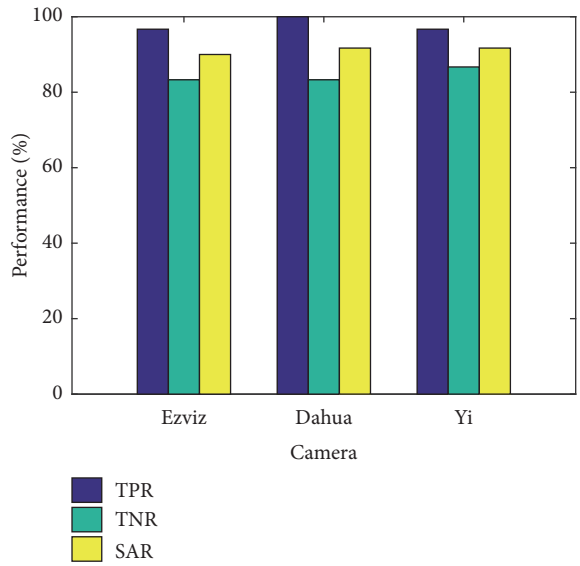


Figure 12: The impact of pets on the user presence inference.

for example, burglary, they shall try to avoid any attention, even that of pets. A break-in may cause a household dog to bark and thus alarm the neighbors. Therefore, from the perspective of the attacker, they shall consider the case of pets and choose a small threshold to achieve a low false negative rate to reduce the chance of being caught.

6. Discussion

With the elaborate design, HomeSpy achieves high performance most of the time. However, HomeSpy has the following limitations.

First, HomeSpy is mainly applicable to houses with wireless cameras for home monitoring. For houses with no cameras, or with local-storage cameras or wired cameras, HomeSpy is inapplicable. However, considering the booming market of wireless cameras, we believe HomeSpy can attack a large number of houses and infer their user presence status.

Second, HomeSpy targets nonintensive houses. For intensive residences, for example, apartments, HomeSpy needs to further determine whether the detected wireless camera is inside the target apartment or not. However, we assume it can be achieved by widely studied wireless localization techniques [18–22] and keep it as the further work.

Third, HomeSpy is designed based on the general principle of wireless cameras in today's market, that is, compression and fragmentation on video and audio streams. If the camera working mode changes dramatically in the future, HomeSpy may not function well and should be updated to incorporate the new working mode accordingly.

Fourth, HomeSpy utilizes the human intervention (i.e., motion and sound) reflected on the camera traffic to infer the user's presence. Thus, HomeSpy works in the scenario where limited motion is performed, but sounds exist, for example, watching movies. However, if the user is not within the filming range of the camera or displays both limited motion and sound (e.g., sleeping or reading), they may not cause enough variation on the camera bitrate and, thus, render HomeSpy ineffective.

7. Related Work

Privacy Concerns with Cameras. Much attention has been paid to the privacy issues brought by the widespread cameras. Existing literature safeguards personal privacy against cameras from several aspects. The first type detects hidden cameras to ensure personal privacy. This work [10] detects drones that carry out privacy invasion attacks with on-board cameras by analyzing RSSI signals. A few applications available on both the Android and iOS platforms detect hidden cameras against secret videotaping. They utilize the signs of light reflection caused by the lenses of a hidden wireless camera [24–26], or the electromagnetic waves (produced by the crystal oscillator) emitted by working cameras [27, 28] to detect hidden cameras. This work [9], on the other hand, prevents unauthorized videotaping by introducing an "invisible light beacon" implemented on the eye-wear. Another type of work [5–8] focuses on the privacy concerns caused by "first-person" wearable cameras and proposes methods to identify and prevent sharing of sensitive images captured by wearable cameras.

Similar to [29], HomeSpy is inspired by previous work and reveals the perspective of attackers that the home security camera can be a potential source of privacy leakage, that is, the user presence status.

Traffic Identification. Essentially, HomeSpy is one type of the traffic identification (TI) problem. However, HomeSpy differs from the traditional TI and is more challenging. Traditional TI approaches [30–34] mainly take the 5-tuple information as inputs, while HomeSpy can only use the PHY/MAC layer information without decoding the TCP/IP headers. This makes HomeSpy challenging and applicable to larger types of scenarios.

8. Conclusion

In this paper, we reveal that wireless cameras in smart homes can leak personal privacy that may result in security issues. We propose HomeSpy, a system that is able to infer whether owners are at home or not, by eavesdropping the traffic of wireless surveillance cameras. We implement HomeSpy on the Android platform and validate it on 20 cameras. The evaluation results show that HomeSpy can achieve a successful attack rate of 97.2%.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work has been funded in part by NSFC 61472358, NSFC 61702451, and the Fundamental Research Funds for the Central Universities 2017QNA4017.

References

[1] Technavio, "Global video surveillance market 2016-2020," http://www.technavio.com/report/global-it-security-video-surveillance-market, 2015.

[2] R. Kenner, "Man sues after finding hidden cam in hotel bathroom," http://www.rense.com/general29/msn.htm, 2012.

[3] A. Castelan and J. Treanor, "Hidden cameras found inside a las vegas airbnb rental recording naked people," https://tinyurl.com/zrtysgx, 2016.

[4] KpopJoA, "Korean couple discover a hidden camera inside airbnb guesthouse in japan," https://tinyurl.com/y9ju55ut, 2017.

[5] A. Ashok, V. Nguyen, M. Gruteser, N. Mandayam, W. Yuan, and K. Dana, "Do not share! Invisible light beacons for signaling preferences to privacy-respecting cameras," in *Proceedings of the 1st ACM MobiCom Workshop on Visible Light Communication Systems, VLCS 2014*, pp. 39–44, USA, September 2014.

[6] R. Templeman, M. Korayem, D. Crandall, and A. Kapadia, "PlaceAvoider: steering first-person cameras away from sensitive spaces," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, USA, 2014.

[7] R. Hoyle, R. Templeman, S. Armes, D. Anthony, D. Crandall, and A. Kapadia, "Pprivacy behaviors of lifeloggers using wearable cameras," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp 2014*, pp. 571–582, USA, September 2014.

[8] R. Hoyle, R. Templeman, D. Anthony, D. Crandall, and A. Kapadia, "Sensitive lifelogs," in *Proceedings of the the 33rd Annual ACM Conference*, pp. 1645–1648, Seoul, Republic of Korea, April 2015.

[9] M. Korayem, R. Templeman, D. Chen, D. Crandall, and A. Kapadia, "Enhancing lifelogging privacy by detecting screens," in *Proceedings of the the 2016 CHI Conference*, pp. 4309–4314, Santa Clara, Calif, USA, May 2016.

[10] S. Birnbach, R. Baker, and I. Martinovic, "Wi-Fly?: detecting privacy invasion attacks by consumer drones," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, USA, 2017.

[11] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[12] M. Bosi, K. Brandenburg, S. Quackenbush et al., "ISO/IEC MPEG-2 advanced audio coding," *Journal of the Audio Engineering Society*, vol. 45, no. 10, pp. 789–814, 1997.

[13] M. Jiang, X. Yi, and N. Ling, "Improved frame-layer rate control for H.264 using MAD ratio," in *Proceedings of the 2004 IEEE International Symposium on Circuits and Systems*, pp. 813–816, Vancouver, BC, Canada.

[14] W. Group, "Amendment 6: Medium access control (mac) security enhancements," *IEEE Standard Association*, vol. 802, 2004.

[15] A. H. Lashkari, M. M. S. Danesh, and B. Samadi, "A survey on wireless security protocols (WEP, WPA and WPA2/802.11i)," in *Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT '09)*, pp. 48–52, IEEE, Beijing, China, August 2009.

[16] M. Schulz, D. Wegemer, and M. Hollick, "Nexmon: The c-based firmware patching framework," https://nexmon.org, 2017.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 37, no. 6, pp. 1067–1080, 2007.

[19] I. Guvenc and C.-C. Chong, "A survey on toa based wireless localization and nlos mitigation techniques," *IEEE Communications Surveys Tutorials*, vol. 11, no. 3, 2009.

[20] R. W. Ouyang, A. K.-S. Wong, and C.-T. Lea, "Received signal strength-based wireless localization via semidefinite programming: noncooperative and cooperative schemes," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1307–1318, 2010.

[21] Z. Farid, R. Nordin, and M. Ismail, "Recent advances in wireless indoor localization techniques and system," *Journal of Computer Networks and Communications*, vol. 2013, Article ID 185138, 12 pages, 2013.

[22] L. Shangguan, Z. Yang, A. X. Liu, Z. Zhou, and Y. Liu, "STPP: spatial-temporal phase profiling-based method for relative RFID tag localization," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 596–609, 2017.

[23] G. Miller, "Cusum control charts," 1993.

[24] A. L. Poretz, "Spy hidden camera detector," https://tinyurl.com/ybkep49w, 2017.

[25] L. LSC, "Hidden camera detector," https://tinyurl.com/jp3xl43, 2016.

[26] Workshop512, "Glint finder-camera detector," https://tinyurl.com/pbnfnur, 2013.

[27] FutureApps, "Hidden camera detector," https://tinyurl.com/y7nlhbx8, 2017.

[28] GalaxyApp, "Hidden camera detector pro," https://tinyurl.com/y745nqf7, 2016.

[29] Y. Cheng, X. Ji, X. Zhou, and W. Xu, "HomeSpy: inferring user presence via encrypted traffic of home surveillance camera," in *Proceedings of the 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 779–782, Shenzhen, 2017.

[30] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel traffic classification in the dark," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 229–240, 2005.

[31] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, p. 50, 2005.

[32] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing skype traffic: When randomness plays with you," in *Proceedings of the ACM SIGCOMM 2007: Conference on Computer Communications*, pp. 37–48, Japan, August 2007.

[33] M. Iliofotou, B. Gallagher, T. Eliassi-Rad, G. Xie, and M. Faloutsos, "Profiling-by-association: a resilient traffic profiling solution for the Internet backbone," in *Proceedings of the 6th International Conference on Emerging Networking Experiments and Technologies (Co-NEXT '10)*, Philadelphia, Pa, USA, December 2010.

[34] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and Y. Xiang, "Internet traffic classification by aggregating correlated naive bayes predictions," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 5–15, 2013.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 5103270, 11 pages
https://doi.org/10.1155/2018/5103270

Research Article

Function-Aware Anomaly Detection Based on Wavelet Neural Network for Industrial Control Communication

Ming Wan[1], **Yan Song**[2], **Yuan Jing**[1] and **Junlu Wang**[1]

[1]*School of Information, Liaoning University, Shenyang 110036, China*
[2]*School of Physics, Liaoning University, Shenyang 110036, China*

Correspondence should be addressed to Yan Song; song.yan@lnu.edu.cn

Received 31 May 2018; Revised 15 August 2018; Accepted 28 August 2018; Published 12 September 2018

Academic Editor: Chunqiang Hu

Function control, which is an essential link in industrial automation, is undergoing a growing integration with ICTs (Information Communication Technologies) because of the flexible manufacturing and convenient interoperability in CPSs (Cyber-Physical Systems). However, it has also brought the increasing dangers of cyberattacks caused by malicious or intentional industrial process control exploitations. In order to effectively detect these cyber intrusions and anomalies, this paper proposes a function-aware anomaly detection approach based on WNN (Wavelet Neural Network), which perceives the abnormal function control changes in industrial control communication. By appropriately extracting the time-related function control characteristics from industrial communication packets, this approach builds an optimized wavelet neural network to model the normal function control behaviors and calculates the detection threshold to differentiate the aberrant industrial process control activities. Additionally, a real-world control system, whose communication protocol is Modbus/TCP, is simulated to furnish the analyzed function control data. According to the experimental results, we fully demonstrate this approach has the fine detection accuracy and adequate real-time capability.

1. Introduction

Nowadays, almost all CPSs (Cyber-Physical Systems) in critical infrastructures (such as electrical and petrochemical systems, sewage systems, and transportation systems) concerning the national economy and the people's livelihood have developed industrial control systems to realize significant automation of industrial processes [1, 2]. In particular, with the rise of Industry 4.0 and Internet of Things [3, 4], the flexible manufacturing and convenient interoperability has already been brought into schedule by academia and industry. Actually, smart CPSs can unleash strong driving forces for the innovation and integration of industrialization and informatization. As an applicable solution, information communication technologies have a positive influence on strengthening traditional industrial control systems [5]. However, the application of ICTs is gradually breaking the original "information island" status of industrial control systems, and the incoming cybersecurity can be dramatically impacted. In consequence, many experienced engineers shift their focus from the process safety to the information security [6]. Over the past several years, CPSs came under cyberattacks from all sides. According to the ICS-CERT (Industrial Control Systems Cyber Emergency Response Team) statistics [7], the ICS-CERT incident response team generalized and analyzed 290 industrial security incidents in 2016, and more and more sophisticated attacks against industrial control systems are developed by the adversaries. Actually, three comprehensible causes in such a situation can be recognized as follows: (1) multifarious vulnerabilities of industrial control systems have been exposed gradually in recent years, for example, system architecture vulnerability [8, 9], embedded control device vulnerability [9–11], and industrial communication vulnerability [11, 12]; (2) the types of cyberattacks are distinctive and diversified, and targeted attacks and APTs (Advanced Persistent Threats) have permeated to face reality [13]; (3) industrial-oriented defense technologies are in an underway and exploring stage, and the regular Internet security methods are unable to satisfy the special industrial control requirements [14].

Actually, the basic industrial control operations of existing DCS (Distributed Control System), SCADA (Supervisory Control And Data Acquisition), and PLC (Programmable Logic Controller) range over two aspects: function control and data acquisition [8, 15]. As an essential link in industrial process automation, function control always performs a series of well-organized and synergetic operations in industrial production and manufacture. For instance, in existing automobile manufacturing process, industrial robots under the master controls can complete the assembly of automobiles according to the predetermined procedures. Therefore, once some attacker deliberately destroys or disturbs the function control process by cyberattacks, huge losses may be caused.

In order to combat this tendency, the researchers make a preliminary probe into two types of defense methods: device-oriented and network-oriented cases. In the device-oriented cases, trusted computing for industrial embedded devices [16] is a burgeoning security technology to provide system integrity check and data confidentiality protection. In the network-oriented cases, industrial firewall [11, 17] and intrusion detection [15, 18, 19] are the typical applications in industrial control networks to improve the communication security. However, because we have not understood the boundary conditions between the availability and security of industrial control systems, the cases on trusted computing and industrial firewall may result in the processing delay or transmission delay in industrial process automation. In general, intrusion detection can discover potential intrusion behaviors in real time by collecting and analyzing various industrial network data, and it scarcely has an impact on industrial control communication activities due to the inconspicuous network sniffer. Moreover, intrusion detection in industrial control systems mainly includes signature-based approaches and anomaly-based approaches, and anomaly-based approaches can detect some unknown attacks by automatically classifying significant deviations from a learned normal behavior model [1]. However, two particular problems should deserve careful considerations: (1) extract the appropriate features according to industrial communication characteristics; (2) build an optimal detection model suited for the extracted features.

In this paper, we propose a function-aware anomaly detection approach based on WNN (Wavelet Neural Network) to identify industrial communication intrusions or anomalies. In particular, these intrusions or anomalies may cause the function control changes in industrial control communication. Furthermore, our approach extracts the time-related features from the communication packets to describe the function control characteristics and build an optimal behavior model based on WNN by using the normal function control samples. With the establishment of behavior model, the detection threshold can be calculated as a scale to differentiate the aberrant industrial control communication activities. Finally, a real-world control system is simulated to furnish the analyzed function control data, and the used industrial communication protocol in this system is Modbus/TCP. According to the experimental results, we fully demonstrate that this approach has the fine detection accuracy and adequate real-time capability.

The major contributions and advantages of this paper involve three aspects: Firstly, we propose a novel time-related feature calculation and construction algorithm to adequately describe the function control characteristics, and this algorithm can slickly extract function control behaviors from industrial control communication activities. Secondly, based on the time-related function control behaviors, we introduce the optimized wavelet neural network to realize the function-ware anomaly detection. Finally, a real-world control system is simulated to evaluate our approach, and the experimental results show that our approach is practicable and effective. Actually, the biggest difference of our approach focuses on the first aspect. That is, adequately modeling function control behaviors is one necessary prerequisite to further explore the real-time anomaly detection. In our approach, we design an original function control feature calculation and construction algorithm to overcome this difficulty.

2. Related Work

According to different detection techniques, the anomaly detection approaches in CPSs can involve three major aspects: rule matching, statistics analysis, and computational intelligence [18]. In the rule matching ones, the prior knowledge must be prepared to learn the general rules for intrusion detection and the rule match is executed to detect many kinds of attacks. Typically, Almalawi et al. [20] automatically extract the proximity detection rules from the consistent and inconsistent states of SCADA data to identify integrity attacks on SCADA systems. Genge et al. [21] propose a systematic and auto-configured anomaly detection approach, which includes modeling of ICS networks and generating anomaly detection rules, to identify the attacks violating ICS connection patterns. Due to the predefined rules, these approaches can improve the classification accuracy and have the practical detection efficiency. But the huge rule database is hard to build and update, because the extracted rules must cover all known attack instances. Besides, they also lack the ability to exploit the unknown attacks which frequently occur in today's CPSs. In the statistics analysis ones, the underlying distribution (such as the network traffic profile) can be learned to detect anomalies, and these techniques are better able to resist the incomplete and imprecise training data than the rule matching ones. For instance, Do [22] and Gawand et al. [23] introduce the CUSUM mechanism to detect the change point of industrial communication traffic. Different from the rule matching ones, these approaches can attempt to find the weaknesses of the unknown attacks, but this ability is very limited because the sophisticated and targeted attacks can easily bypass the distribution changes. Moreover, the high false positive and negative rate is another drawback because it is difficult to determine the traffic profile. In the computational intelligence-based ones, these techniques always have a strong correlation with data mining. Furthermore, the normal models or profiles are built from multivariate training data, and the corresponding anomaly detection is realized by using the mechanism of classification or optimization. Actually, the computational intelligence techniques have been attracting great interests of both industry and academia, and

many computational intelligence approaches have been researched, mainly including SVM (Support Vector Method) [15, 24, 25], neural network [26], decision trees [27], genetic algorithm [27, 28], and clustering technique [29]. Although the computational intelligence-based techniques have the relatively high computational overhead, they can achieve better performance in detection, tolerance, and generality [14]. Additionally, these approaches can not only detect known attacks with high detection efficiency, but also have a better function in identifying new intrusion modes [15]. It is worth mentioning that our approach belongs to the computational intelligence ones. Differently, we propose a new feature calculation and construction algorithm for industrial control communication, which not only successfully extracts the function control behavior from industrial communication characteristics, but also moderately reduces the computational complexity.

3. Function Control Feature Calculation and Construction

In industrial control communication, function codes, which represent control signals sent from the operator or engineer workstations, are distributed to the executive devices for the purpose of controlling industrial automation process. Therefore, the feature calculation and construction algorithm analyzes the time-related function codes to simulate the function control behavior. In particular, we cannot simply gather the function codes at regular time intervals as function control samples to train the behavior model, and the intrinsic reasons include the following: (1) the number of function codes at each regular time interval is distinct, and the prerequisite for the behavior model based on WNN is that the dimensions of input samples must be consistent with one another; (2) the number of function codes at each regular time interval may be very large, and it may waste computational resources and reduce detection efficiency. Figure 1 depicts the detailed feature calculation and construction process, and each step can be outlined below.

Step 1 (function code sequence preprocessing). Like our prior work in [15], in order to associate time characteristics with function control activities, we first parse the captured function control packets in depth and obtain the function code sequence $F_i = f_1^i f_2^i f_3^i \cdots f_{m_i}^i$ in every interval t (here, m_i is the serial number of every function code in F_i). The function code sequence set $F_{set} = \{F_1, F_2, \cdots, F_n\}$ in the interval T ($T = \sum_{i=1}^{n} t$) can consist of all function code sequences F_i ($i = 1, \cdots, n$), and all sequence dimensions in F_{set} can separate from each other because of the different m_i in each sequence. After that, we recombine all F_i ($i = 1, \cdots, n$) to the one big sequence F according to the time order.

Step 2 (feature factor selection). Because the dimensions of obtained function control samples X_i ($i = 1, \cdots, n$) must be consistent, we first need to construct the feature base vector according to the selected feature factors. In particular, the selected feature factors consist of two main components: single function code and short sequence pattern. More specifically, all single function codes $f_1, f_2, \cdots, f_v$ are searched in sequential order from the large function code sequence F, and each single function code is different from the others. According to each single function code f_j ($j \in [1, v]$), we design the short sequence pattern M_k^j ($k \in [1, v]$). Furthermore, M_k^j consists of f_j and f_k, and we can get v short sequence patterns $M_1^j, M_2^j, \cdots, M_v^j$ for each single function code f_j. Taken together, we can acquire the feature base vector $f_1, M_1^1, M_2^1, \ldots, M_v^1, \cdots, f_v, M_1^v, M_2^v, \ldots, M_v^v$. The intrinsic reasons for such feature factor selection include the following: (1) the single function code can represent its own role in each function code sequence; (2) the short sequence pattern can establish the relationship between two function codes and indirectly reflect the continuous control operations in industrial automation process.

Step 3 (function control sample calculation). According to the feature base vector, we further calculate the corresponding feature variable for each feature factor in the function code sequence F_i. For the single function code f_j, we regard its frequency x_j in F_i as the corresponding feature variable, and the calculation formula is $x_j = g(f_j)/m_i$; here $g(f_j)$ represents the number of f_j in F_i. For the short sequence pattern M_k^j, we calculate its frequency x_k^j in F_i as the corresponding feature variable by the formula $x_k^j = g(M_k^j)/(m_i - 1)$. By calculating all feature variables in F_i, we can complete the construction of the sample $X_i = (x_1, x_1^1, x_2^1, \ldots, x_v^1, \cdots, x_v, x_1^v, x_2^v, \ldots, x_v^v)$. Additionally, there is a one-to-one correspondence between function code sequences and function control sample, and each function control sample contains $v(1 + v)$ feature variables. To sum up, all calculated function control samples form the function control sample set $X_{set} = \{X_1, X_2, \cdots, X_n\}$.

4. Function-Aware Anomaly Detection Based on Wavelet Neural Network

After the feature construction, we can train a wavelet neural network to discover any functional change in industrial control communication. Moreover, we introduce the WNN's prediction capability to realize that the function-aware anomaly detection, an optimized WNN, and the correlative detection threshold are achieved by the loop-based iterative train.

4.1. Architecture Design. Figure 2 shows the overall architectural design of function-aware anomaly detection based on WNN. As this figure shows, this detection approach is made up of two phases: model training and real-time detection. Actually, model training is an essential step or a prerequisite in order to improve the detection accuracy. In this phase, by using the training function control samples extracted from the normal industrial communication data, an optimized WNN-based behavior model is successfully built and an accompanying detection threshold (including an upper limit and a lower limit) is measured and recorded.

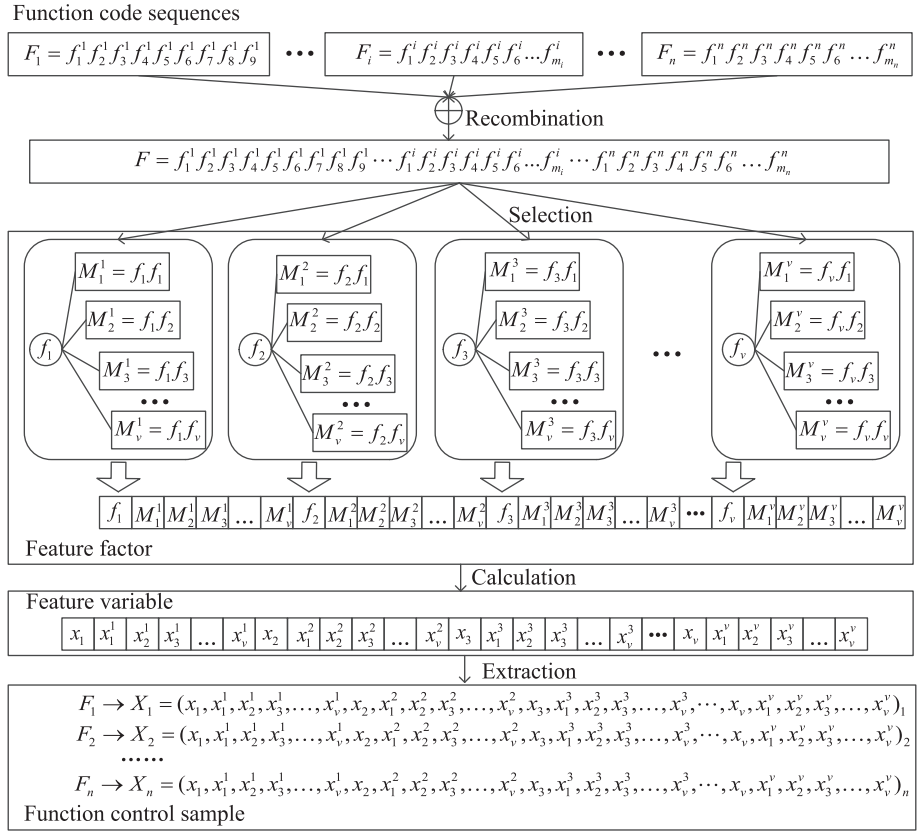


Figure 1: Detailed feature calculation and construction process according to function control characteristics.

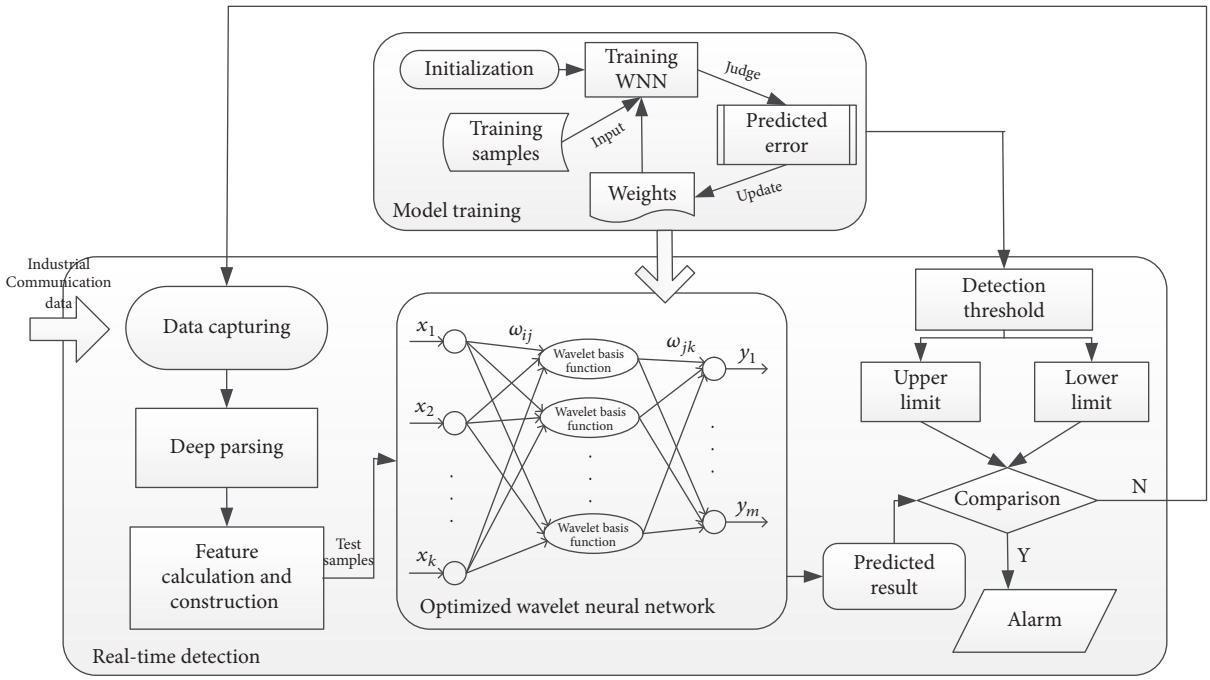


Figure 2: Overall architectural design of function-ware anomaly detection approach based on WNN.

In the real-time detection phase, industrial communication data are captured and parsed in depth to form the test function control samples by means of the feature calculation and construction algorithm mentioned in Section 3, and the optimized wavelet neural network analyzes these input test samples to calculate the predicted results, which are further compared with the detection threshold. When the predicted results are not covered by the detection threshold, an alarm will be generated in real-time.

4.2. Wavelet Neural Network and Optimization. WNN has already been successfully applied to many practical areas [30], and in our approach it is introduced as the critical behavior model to identify function control misbehaviors. In practice, the topological structure of WNN evolves from BP neural network, and it regards the wavelet basis function as the activation function of hidden layer wavelons, which are referred to as the hidden units. In the hidden layer, the input variables are inserted and transformed to wavelets, and all wavelons are combined to estimate the approximation of the target values [31].

In the WNN's structure depicted in Figure 2, $x_1, x_2, \cdots, x_k$ are the input variables in the input layer, and $y_1, y_2, \cdots, y_m$ are the predicted results in the output layer. Additionally, ω_{ij} and ω_{jk} stand for the network weights. If the input variables are x_i $(i = 1, 2, \cdots, k)$, the corresponding outputs can be given by the expression:

$$h(j) = h_j\left(\frac{g(x) - b_j}{a_j}\right)$$
$$g(x) = \sum_{i=1}^{k} \omega_{ij} x_i, \quad j = 1, 2, \cdots, l \tag{1}$$

Here, $h(j)$ is the output of the hidden unit j in the hidden layer; ω_{ij} is the connection weight between the input layer and the hidden layer; b_j represents the translation parameter of the wavelet basis function h_j; a_j represents the dilation parameter of the wavelet basis function h_j.

In our wavelet neural network, the Morlet wavelet is selected as the wavelet basis function, given by

$$y = \cos(1.75x)\, e^{-x^2/2} \tag{2}$$

After the calculation of the hidden layer, we can further obtain the predicted results by the following expression:

$$y(k) = \sum_{j=1}^{l} \omega_{jk} h(j), \quad k = 1, 2, \cdots, m \tag{3}$$

Here, ω_{jk} is the connection weight between the hidden layer and the output layer; $h(i)$ is the output of the hidden unit i in the hidden layer; l is the number of the hidden units; m is the number of the output units.

It is worth mentioning that we use the loop-based iteration to train the optimized WNN, and its main purpose is to improve the network parameters, including the connection weights ω_{ij} and ω_{jk}, the translation parameter b_j, and the dilation parameter a_j. Furthermore, the predicted error is introduced to shorten the distance between the predicted results and the expected outputs, and the predicted error can be computed by

$$e_p = \sum_{k=1}^{m} \left[y'(k) - y(k)\right] \tag{4}$$

Here, $y'(k)$ is the expected output, and $y(k)$ is the predicted result.

Algorithm 1 shows the pseudocode of WNN's optimization process. In this process, the parameter increments are introduced to update all network parameters, and the specific process can refer to the WNN's training steps in Section 4.3. In practice, two different terminations of iteration for this process can be selected: one is the maximum number of iterations, and the other is the preconfigured error threshold which indicates the iteration is completed if the distance between the predicted results and the expected outputs is small enough. In our approach, we select the first one as the terminal condition.

4.3. Training and Detection. As mentioned in Section 4.1, the main steps of model training are outlined below.

Step 1 (network parameter initialization). We first initialize the primary parameters of wavelet neural network, including the dilation parameter a_j, the translation parameter b_j, and the connection weights ω_{ij} and ω_{jk}. Additionally, we also set the learning rate, which is used to improve the above parameters.

Step 2 (predicted error calculation). According the training function control samples, we calculate the predicted error by (4).

Step 3 (parameter modification). On the basis of the predicted error, we further improve the network parameters to shorten the distance between the predicted results and the expected outputs.

Step 4 (detection threshold measurement). Finally, we repeat Steps 2 and 3 until the iteration ended and record the optimized detection threshold.

After the model training, we can perform the real-time detection to identify function control misbehaviors. Moreover, the basic prerequisite is that we must resolve the real-time function control samples from the observed industrial communication data by using our feature calculation and construction algorithm. As the input variables, these samples can further be analyzed by the optimized wavelet neural network to estimate the predicted results, which will be compared with the detection threshold. The judgment criterion to generate an alarm is that if the predicted results fall within the range from the lower limit to the upper limit, we can believe these function control activities are normal; if the

Do initializing network parameters ($\omega_{ij}, \omega_{jk}, a_j, b_j$)
Do initializing parameter increments ($\Delta\omega_{ij}, \Delta\omega_{jk}, \Delta a_j, \Delta b_j$)
Do setting iteration number *IntNum*
for *1* to *IntNum*
  for all input variables
    for *1* to *l*
      Do computing $h(j)$ and $g(x)$ by *Eq. (1)*
      Do computing $y(k)$ by *Eq. (3)*
    end
    Do computing and recording predicted error e_p by *Eq. (4)*
    for *1* to *l*
      Do correcting parameter increment ($\Delta\omega_{ij}, \Delta\omega_{jk}, \Delta a_j, \Delta b_j$)
    end
    Do updating connection weights ω_{ij} and ω_{jk} by $\Delta\omega_{ij}$ and $\Delta\omega_{jk}$
    Do updating dilation parameter a_j by Δa_j
    Do updating translation parameter b_j by Δb_j
  end
end

Algorithm 1: Pseudocode of WNN's optimization process.

predicted results escape from these ranges, we may doubt the corresponding function control activities are abnormal.

5. Experimental Analysis and Discussion

5.1. Experimental Modbus/TCP Control System. In order to evaluate the detection performance, we use the simulation control system which is built in our earlier work [15] to furnish the analyzed function control data. Furthermore, the industrial control communication of this system is based on Modbus/TCP, in which various function codes are utilized to facilitate different control operations. Figure 3 shows the basic network architecture of this control system. Furthermore, the chief purpose of this system is to accomplish the material production by monitoring and controlling the valves and the liquid levels, and the detailed technological process has been presented in [15]. In particular, the whole technological process is repeated every 1 minute. Besides, in this control system we carry out some attack experiments to forge and replay some malicious Modbus control commands, and our ultimate goal is to evaluate the detection accuracy and real-time capability by using these malicious function control data.

The normal communication packets are captured from the industrial switch to train the optimized wavelet neural network, and the capture time lasts 1h15m02s. After the preliminary statistics, the number of Modbus/TCP function codes in these packets reaches 11693. Additionally, we also use Matlab to analyze these packets in depth, and the hardware configurations are also the same with the ones in [15]. Per one minute, we compute the number of different function codes, and all statistical results are shown in Figure 4. As these results show, the simulation control system uses four categories of function codes to complete the whole technological process, and these function codes are 1, 3, 5, and 6, respectively. Besides, all five curves in this figure flatten out, and the number of every function code fluctuates smoothly. In brief, these results can also demonstrate that the simulation control system has the relatively steady communication patterns under normal circumstances, and its function control status appears on a relatively limited range.

5.2. Detection Performance Evaluation. Without loss of generality, we choose the detection accuracy and real-time capability as the main performance indicators to evaluate our approach. Before training the optimal behavior model, we first preprocess the captured Modbus/TCP packets. More specifically, we extract the function codes per 1 minute to form the function code sequences, and by using the feature calculation and construction algorithm we win a total of 75 function control samples. Because 4 function codes exist in the simulated technological process, each function control sample contains 20 feature variables. According to these normal function control samples, we further train and optimize the wavelet neural network. It is worth mentioning that we set the number of iterations to 200 in order to reduce the predicted error, and Figure 5 plots the change curve of predicted errors with the iteration times. From this figure we can see that, along with the increasing of iteration times, the curve of predicted errors changes from rapid reduction to gentle trend. In particular, the best detection accuracy in the 200th iteration can reach 98.67%; that is, the predicted accuracy of the optimized WNN to detect 75 normal function control samples can reach 98.67%, and only one normal function control sample is mistakenly regarded as the outlier.

By using the optimal behavior model, we further evaluate its detection performance, including detection accuracy and consuming time. In each experiment, we forge and replay some malicious Modbus/TCP packets to attack and destroy the normal technological process. Moreover, we suppose that these malicious Modus/TCP packets cannot contain other function codes which are different with the four categories

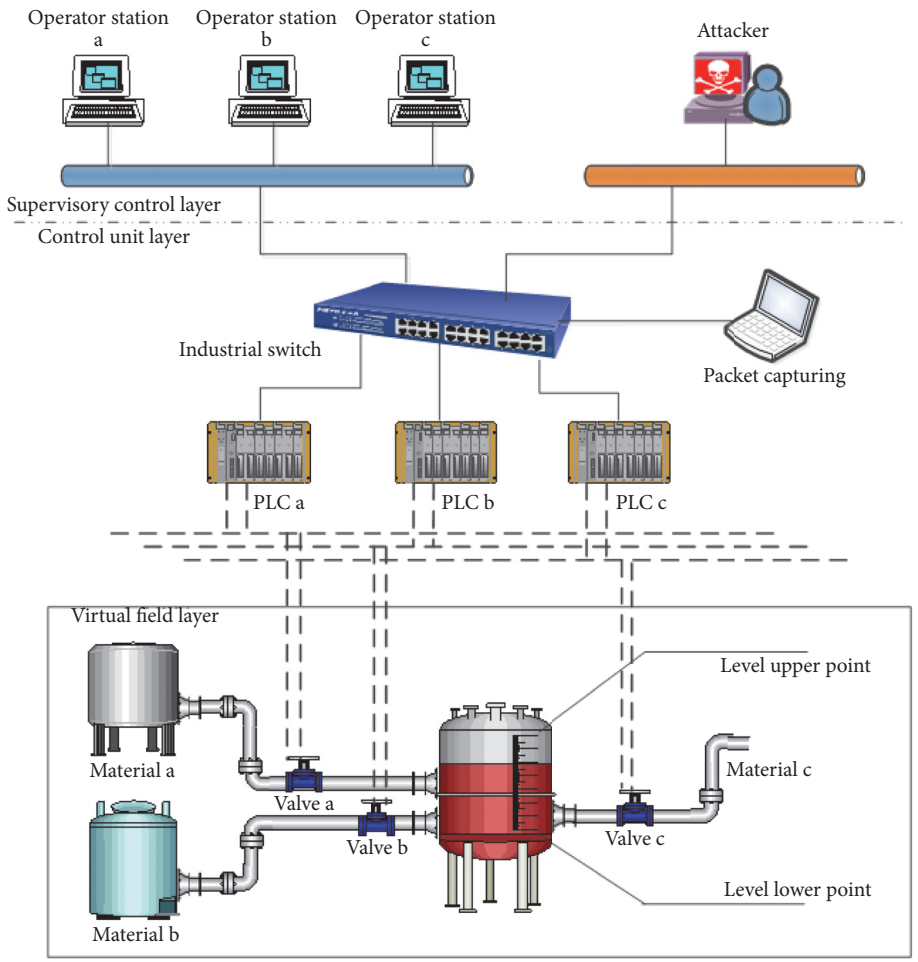


Figure 3: Basic network architecture of simulated Modbus/TCP control system [15].

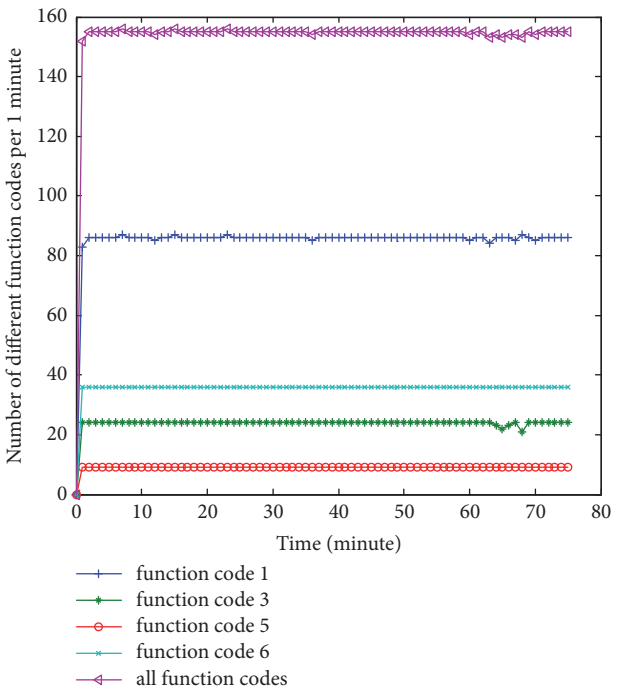


Figure 4: Statistical results of different function code number.

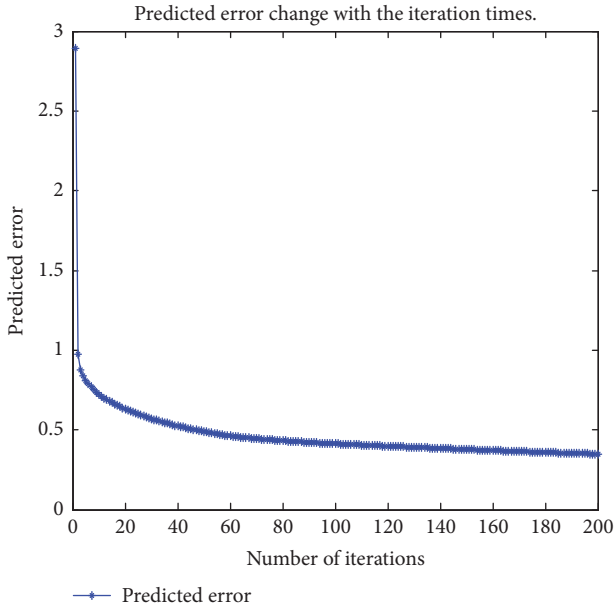


Figure 5: Predicted error change with the iteration times.

of function codes used in the simulation control system, and these packets only change the function control process. The major reason of such assumption is that the malicious packets containing other function codes can be easily filtered by the applied industrial firewall [11, 32]. Besides, we generate 60 malicious function code sequences in each experiment. More specially, the percentage of the malicious function codes in each function code sequence is about 1/10, and the locations of the malicious function codes in each function code sequence can be considered random. Similarly, we can obtain 60 malicious function control samples in each experiment after the feature calculation and construction. By calculating the predicted result for each malicious function control sample, we compare it with the detection threshold to identify the corresponding abnormal function control behavior. Table 1 shows the experimental results of detection performance under 10 different experiments in detail. In this table, the average detection accuracy is 91.17%, and the average consuming time is 0.0104s. In the extreme case, the smallest detection accuracy is 88.33% in the 3rd, 7th, and 10th experiments, and the largest consuming time is only 0.0281s to detect 60 function control samples in the 7th experiment. In a word, we fully demonstrate the function-ware anomaly detection approach has the fine detection accuracy and adequate real-time capability; namely, they indirectly declare it has the remarkable capacity to differentiate the abnormal function control activities.

Actually, the adversary can change the attack intensity by adjusting the attack frequency; for example, the sending rate of malicious Modbus/TCP packets can be increased by the adversary to launch an attack with a higher probability of success. Therefore, the percentage of the malicious function codes in each function code sequence may also change accordingly. However, the different percentages of the malicious function codes in each function code sequence

Table 1: Detection accuracy and consuming time of 10 experiments.

	Detection accuracy	Consuming time
1	90.00%	0.0066s
2	91.67%	0.0101s
3	88.33%	0.0072s
4	93.33%	0.0068s
5	90.00%	0.0066s
6	95.00%	0.0075s
7	88.33%	0.0281s
8	91.67%	0.0127s
9	95.00%	0.0099s
10	88.33%	0.0084s
Average value	91.17%	0.0104s

can have a marked impact on the detection accuracy of our approach. In order to define different influencing effects, we suppose the percentages of the malicious function codes are 1/5, 1/10, 1/15, 1/20, 1/25, and 1/30, and 5 distinct experiments are performed for each percentage. Similarly, we also generate 60 malicious function control samples in each experiment. Figure 6 plots the detection accuracy variation under different percentages of the malicious function codes in each function code sequence. In this figure, $p_1, p_2, \cdots, p_6$ represent the percentages whose values are 1/5, 1/10, 1/15, 1/20, 1/25, and 1/30, respectively, and the minimum detection accuracies, the average detection accuracies, and the maximum detection accuracies are plotted according to every 5 experiments. Viewed generally, the experimental results reflect the detection accuracy also decreases with the reduction of the percentage; that is, our approach can be more effective in detecting the function control misbehavior caused by the larger percentage of the malicious function codes. However, our approach still maintains a high detection accuracy; for instance, when the percentage is 1/30, the average detection accuracy can reach 76.67%. Additionally, Figure 7 shows the average consuming time under different percentages of the malicious function codes. From this figure, we can see that the consuming time fluctuates remarkably in a narrow range. In other words, the different percentages have almost no influence on the consuming time.

5.3. Compared Analysis. In practice, the innovations of our approach mainly include two aspects: (1) we propose a new feature calculation and construction algorithm to extract function control characteristics in industrial control communication; (2) according to the extracted function control samples, we introduce the optimal function-aware WNN model to differentiate the aberrant industrial control communication activities. Therefore, we also provide the compared analysis to explain its advantages from these two aspects.

For one thing, compared with the work in [15, 25], the feature calculation and construction algorithm in this paper can learn more information about function control characteristics from industrial communication packets. On the one hand, this algorithm selects the single function code

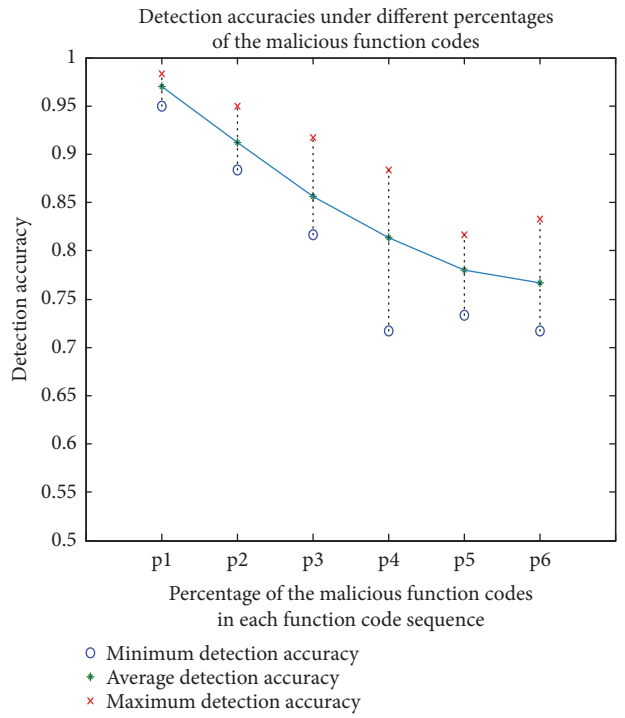


Figure 6: Detection accuracies under different percentages of the malicious function codes in each function code sequence.

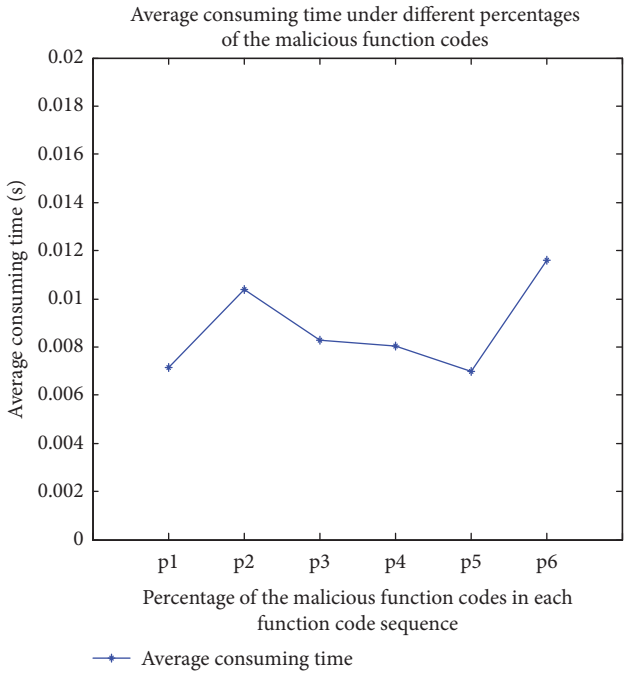


Figure 7: Average consuming time under different percentages of the malicious function codes in each function code sequence.

as an independent feature factor to enhance its own role effect in each function code sequence. On the other hand, the short sequence patterns include all adjacent cases of two function codes, and they not only indirectly reflect the continuous control operations in the normal technological process, but also consider the impact of two nonadjacent control operations in the actual technological process. Therefore, more information about function control characteristics can be utilized to improve the detection efficiency.

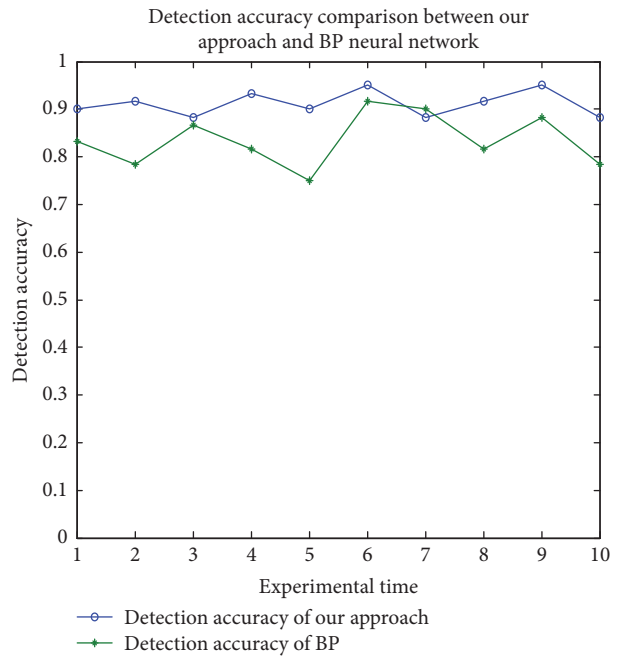


Figure 8: Detection accuracy comparison between our approach and BP neural network under 10 experiments.

Table 2: Average detection accuracies of our approach and BP neural network.

Average detection accuracy	
Our approach	BP neural network
91.17%	83.50%

Based on the same sample extraction by using the proposed feature calculation and construction algorithm, we compare our approach with BP neural network to evaluate the detection accuracy and explain that the proposed approach is more suitable and applicable to detect function control misbehaviors. Similarly, we also perform 10 experiments, and the function control samples in each experiment are the same with the ones whose percentage of the malicious function codes in each function code sequence is about 1/10. Figure 8 plots the detection accuracy comparison between our approach and BP neural network under 10 experiments, and Table 2 shows the corresponding average detection accuracies of two approaches. From these results we can see that BP neural network has a relatively large fluctuation of the detection accuracy, and its average detection accuracy is only 83.50% which is lower than the one of our approach. Therefore, our approach has the ability to provide the better detection accuracy.

6. Conclusion

Aiming at differentiating the aberrant industrial control communication activities, this paper proposes a function-aware anomaly detection approach based on WNN. Firstly, we design the feature calculation and construction algorithm to learn the function control characteristics and extract the time-related features. Secondly, a behavior model based on WNN is established and optimized to detect function control misbehaviors in industrial control communication. Finally, in order to evaluate our approach, we simulate a real-world control system based on Modbus/TCP to perform plenty of experiments, and the experimental results and the compared analysis are offered to express the advantages: our approach has the fine detection accuracy and adequate real-time capability.

Data Availability

In this manuscript, the analyzed function code data are captured and analyzed from our simulation control system, which is built to accomplish the material production according to one real-world control system. Actually, we have sketched the basic technological process in this manuscript, but some contents and specific parameters of this process are not completely open to the public due to the commercialized secrets. Therefore, the analyzed function code data used to support the findings of this study are currently under embargo. If other researchers want to verify the results, replicate the analysis, or conduct secondary analyses, please contact with the corresponding author or first author. The requests for the data will be considered by them after a confidentiality agreement.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61501447), Intelligent Manufacturing Project of the Ministry of Industry and Information Technology: Industrial Internet Data Mutual Recognition Research – Low-Power Message Distribution, and the General Project of Scientific Research of Liaoning Provincial Department of Education (LYB201616).

References

[1] C. Zhou, S. Huang, N. Xiong et al., "Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 10, pp. 1345–1360, 2015.

[2] F. Kargl, R. W. Van Der Heijden, H. König, A. Valdes, and M. C. Dacier, "Insights on the security and dependability of industrial control systems," *IEEE Security & Privacy*, vol. 12, no. 6, pp. 75–78, 2014.

[3] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[4] Y. Huo, C. Hu, X. Qi, and T. Jing, "LoDPD: a location difference-based proximity detection protocol for fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1117–1124, 2017.

[5] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. Nai Fovino, and A. Trombetta, "A multidimensional critical state analysis for detecting intrusions in SCADA systems," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 179–186, 2011.

[6] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th International Symposium on Information, Computer and Communications Security (ASIACCS '11)*, pp. 355–366, Hong Kong, China, March 2011.

[7] ICS-CERT, "ICS-CERT year in review 2016," https://ics-cert.us-cert.gov/sites/default/files/Annual_Reports/Year_in_Review_FY2016_Final_S508C.pdf, 2017.

[8] K. Stouffer, J. Falco, and K. Scarfone, "Guide to Industrial Control Systems (ICS) security," *National Institute of Standards and Technology (NIST) Special Publication 800-82*, 2011, http://csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf.

[9] K. Ly and Y. Jin, "Security challenges in CPS and IoT: from end-node to the system," in *Proceedings of the 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 63–68, Pittsburgh, PA, USA, July 2016.

[10] Y. Huo, Y. Tian, L. Ma, X. Cheng, and T. Jing, "Jamming strategies for physical layer security," *IEEE Wireless Communications Magazine*, vol. 25, no. 1, pp. 148–153, 2018.

[11] M. Wan, W. Shang, L. Kong, and P. Zeng, "Content-based deep communication control for networked control system," *Telecommunication Systems*, vol. 65, no. 1, pp. 155–168, 2017.

[12] Z. Drias, A. Serhrouchni, and O. Vogel, "Taxonomy of attacks on industrial control protocols," in *Proceedings of the 2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS)*, pp. 1–6, Paris, France, July 2015.

[13] F. Skopik, I. Friedberg, and R. Fiedler, "Dealing with advanced persistent threats in smart grid ICT networks," in *Proceedings of the 2014 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference, ISGT 2014*, pp. 1–5, Washington DC, USA, May 2014.

[14] L. Neitzel and B. Huba, "Top ten differences between ICS and IT: Cybersecurity Understanding the different needs of ICS and IT system security leads to cooperation and collaboration between historically disconnected camps," *Intech*, vol. 61, no. 5, pp. 1–6, 2014.

[15] M. Wan, W. Shang, and P. Zeng, "Double behavior characteristics for one-class classification anomaly detection in networked control systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3011–3023, 2017.

[16] P. Maene, J. Gotzfried, R. de Clercq, T. Muller, F. Freiling, and I. Verbauwhede, "Hardware-based trusted computing architectures for isolation and attestation," *IEEE Transactions on Computers*, vol. 67, no. 3, pp. 361–374, 2018.

[17] M. Cheminod, L. Durante, L. Seno, and A. Valenzano, "Performance evaluation and modeling of an industrial application-layer firewall," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2159–2170, 2018.

[18] S. Han, M. Xie, H. H. Chen, and Y. Ling, "Intrusion detection in cyber-physical systems: techniques and challenges," *IEEE Systems Journal*, vol. 8, no. 4, pp. 1052–1062, 2014.

[19] T. Cruz, L. Rosa, J. Proenca et al., "A cybersecurity detection framework for supervisory control and data acquisition systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2236–2246, 2016.

[20] A. Almalawi, X. Yu, Z. Tari, A. Fahad, and I. Khalil, "An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems," *Computers & Security*, vol. 46, pp. 94–110, 2014.

[21] B. Genge, D. A. Rusu, and P. Haller, "A connection pattern-based approach to detect network traffic anomalies in critical infrastructures," in *Proceedings of the the Seventh European Workshop*, pp. 1–6, Amsterdam, The Netherlands, April 2014.

[22] V. L. Do, "Statistical detection and isolation of cyber-physical attacks on SCADA systems," in *Proceedings of the IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 3524–3529, Beijing, China, October 2017.

[23] H. Gawand, A. K. Bhattacharjee, and K. Roy, "Control aware techniques for protection of industrial control system," in *Proceedings of the 2014 Annual IEEE India Conference (INDICON)*, pp. 1–6, Pune, India, December 2014.

[24] A. Terai, S. Abe, S. Kojima, Y. Takano, and I. Koshijima, "Cyber-attack detection for industrial control system monitoring with support vector machine based on communication profile," in *Proceedings of the 2017 IEEE European Symposium on Security and Privacy: Workshops (EuroS&PW)*, pp. 132–138, Paris, France, April 2017.

[25] W. Shang, P. Zeng, M. Wan, L. Li, and P. An, "Intrusion detection algorithm based on OCSVM in industrial control system," *Security and Communication Networks*, vol. 9, no. 10, pp. 1040–1049, 2016.

[26] A. F. Shosha, P. Gladyshev, . Shinn-Shyan Wu, and . Chen-Ching Liu, "Detecting cyber intrusions in SCADA networks using multi-agent collaboration," in *Proceedings of the 2011 16th International Conference on Intelligent System Applications to Power Systems (ISAP)*, pp. 1–7, Hersonissos, Greece, September 2011.

[27] J. Hosic, J. Lamps, and D. H. Hart, "Evolving decision trees to detect anomalies in recurrent ICS networks," in *Proceedings of the World Congress on Industrial Control Systems Security, WCICSS 2015*, pp. 50–57, London, UK, December 2015.

[28] J. E. Rubio, C. Alcaraz, R. Roman, and J. Lopez, "Analysis of intrusion detection systems in industrial ecosystems," in *Proceedings of the 14th International Conference on Security and Cryptography*, pp. 116–128, Madrid, Spain, July 2017.

[29] I. Kiss, B. Genge, and P. Haller, "A clustering-based approach to detect cyber attacks in process control systems," in *Proceedings of the IEEE 13th International Conference on Industrial Informatics*, pp. 142–148, Cambridge, UK, July 2015.

[30] A. K. Alexandridis and A. D. Zapranis, "Wavelet neural networks: a practical guide," *Neural Networks*, vol. 42, pp. 1–27, 2013.

[31] X. Yuan, G. Peng, F. Zhang et al., "Analysis of power grid harmonics with wavelet network," in *Proceedings of the 2017 29th Chinese Control And Decision Conference (CCDC)*, pp. 6052–6055, Chongqing, China, May 2017.

[32] M. Cheminod, L. Durante, A. Valenzano, and C. Zunino, "Performance impact of commercial industrial firewalls on networked control systems," in *Proceedings of the 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, Berlin, Germany, September 2016.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 1821804, 9 pages
https://doi.org/10.1155/2018/1821804

Research Article

A Novel Differential Game Model-Based Intrusion Response Strategy in Fog Computing

Xingshuo An, Fuhong Lin, Shenggang Xu, Li Miao, and Chao Gong

School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB), Beijing 100083, China

Correspondence should be addressed to Fuhong Lin; fhlin@ustb.edu.cn

Received 23 May 2018; Accepted 18 July 2018; Published 1 August 2018

Academic Editor: Liran Ma

Fog computing is an emerging network paradigm. Due to its characteristics (e.g., geo-location and constrained resource), fog computing is subject to a broad range of security threats. Intrusion detection system (IDS) is an essential security technology to deal with the security threats in fog computing. We have introduced a fog computing IDS (FC-IDS) framework in our previous work. In this paper, we study the optimal intrusion response strategy in fog computing based on the FC-IDS scheme proposed in our previous work. We postulate the intrusion process in fog computing and describe it with a mathematical model based on differential game theory. According to this model, the optimal response strategy is obtained corresponding to the optimal intrusion strategy. Theoretical analysis and simulation results demonstrate that our security model can effectively stabilize the intrusion frequency of the invaders in fog computing.

1. Introduction

Fog computing is an emerging network model [1]. As shown in Figure 1, fog computing is a three-layer architecture: user device layer, fog node layer, and cloud computing layer. Fog nodes are service nodes located between cloud and users [2]. Fog nodes [3] are geo-distributed, which can provide low latency services for users. The research in this paper is based on this network architecture.

Fog nodes are located at the edge of the network, which is closer to users. The needs of heterogeneous network access and diverse services make fog nodes face more complex and insecure network environment. The traditional network security technology such as physical security technology [4] is difficult to resist the multisource and cross-domain intrusion [5]. It is necessary to research the network security technology suitable for fog computing to deal with new challenges. Intrusion detection system [6] (IDS) is a measure that can provide effective security for fog network [7]. Our previous work [8, 9] has proposed a general IDS framework to protect cloud servers and fog nodes from security threats. One of the functions of IDS is to make corresponding response strategy based on attackers' behaviors. In this framework, intrusion response is the strategy and action for intrusion when the fog node detects the intrusion. Response strategy selection is the most critical problem in intrusion response [10].

In the fog network, the intruder will attack the fog cluster and carry out an invasion process from fog to cloud. Cloud as a management system for fog cluster needs to respond to such intrusion processes. The intruder implements different frequency attacks on fog nodes. The purpose is to successfully bypass the IDS deployed by the fog node, in order to intrude into the system for further intrusion activities. In other words, the intruder's needs maximized his invasion success expectations. For the system, the cloud server's strategy is to set the access forbidding rate to the fog cluster. In addition to dealing with illegal users, fog cluster also needs to serve legal users. The system needs to serve legal users as much as possible. In order to find the optimal strategy of the intruder and the system, we can regard the problem as a game [10], and the intruder and the system are the players of the game.

In view of intrusion response, some researchers use static game method to model and solve. A universal game model is proposed in [11]. An approach named Response and Recovery Engine (RRE) [12] was proposed. RRE was based on Markov game theory. Reference [13] proposed a dynamic intrusion response model based on game theory to assure the incentives

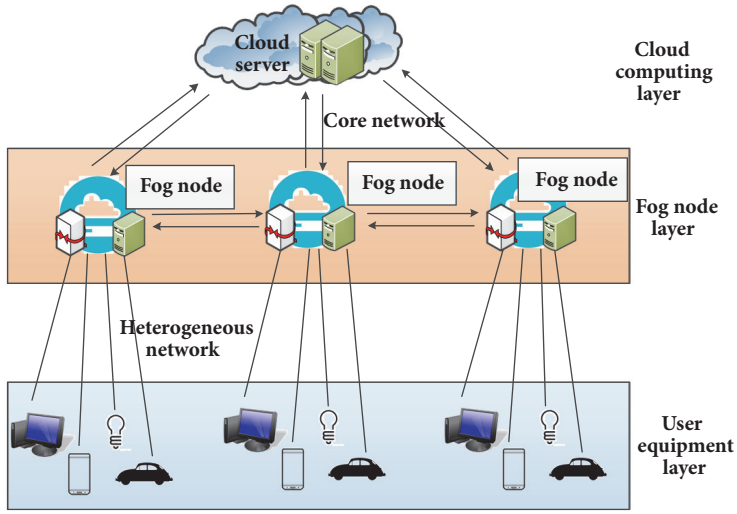


Figure 1: Fog computing network structure.

of system. Noncooperative games model was applied to solve the problem of intrusion response [14]. Using the modeled stochastic game, the authors in [15] proposed a decision working framework to take optimal actions in case of network intrusion.

In fog computing, the interaction between fog nodes and cloud is real time. In continuous time, the cloud needs to make decisions in real time. Accordingly, intruders need to change their strategy in real time to maximize their gains. Differential game [16], as a game model in continuous time, is more suitable for the network environment of fog computing. At present, there is little reference about the application of differential games in the field of fog computing security. The relevant research is only found in [17]. The author defines the strategy of fog nodes from the perspective of energy consumption. The two players of the game are vulnerable node and the malicious nodes in the fog cluster. In this paper, the two players of the game are defined, and strategic analysis from the perspective of the system composed of fog cluster and the perspective of intruders is made, respectively. The main work of this paper is to analyze the characteristics of intrusion in the environment of fog network, apply differential game to model the invaders and system, respectively, and emphasize the theoretical analysis of defense model of the system. In our model, the cloud server can take the best security strategy to filter the access requests of the illegal users based on the attack of invaders. To our knowledge, this is the first differential game theory approach to model the interactions between the intruder and the system in fog computing.

The main contributions of this paper are as follows:

(1) The path and characteristics of invasion are analyzed in the environment of fog computing. The invader model and defense model of the system are built, respectively, according to the invasion.

(2) We derive the optimal strategy of the system and the rational intruder, i.e., the Nash equilibrium of the game.

(3) The simulation shows the outstanding performance of the proposed strategy.

The rest of the paper is organized as follows. In Section 2, we analyzed the intrusion process in fog computing. In Section 3, the differential game models of intruders and system are established and analyzed, respectively. In Section 4, the feedback Nash equilibrium solution is given. The model simulations are provided in Section 5. Finally, the main conclusions are summarized in Section 6.

2. Intrusion in Fog Network

This study focuses on what strategy the system should take when an intrusion occurs. The process that an invasion starts from the fog nodes to the cloud is given. The ultimate goal of invaders is to gain higher permissions on cloud servers, thus causing greater damage to the entire fog network. From the perspective of network attack, invasions from user device layer to fog node layer and then to cloud servers are implemented through different invasion methods. Figure 2 shows the invasion process mentioned above.

Fog nodes are faced with the heterogeneous network environment and communication protocols, and operating system and program bugs are easy to be exploited by invaders. By detecting fog bugs, invaders can find bug in fog nodes. In this process, an invader needs to send a number of access requests to each fog node to detect the bug. When the number of requests sent is too large, it will also cause a denial of service attack (DoS) to the fog node. When an invader finds an available bug, he will exploit bugs to achieve illegal invasions. Once an invader fatally invades a fog node, it will first cause serious harm to the users in the service range of the fog node, such as Privacy leakage and Malware propagation. Secondly, invaders will directly have a negative impact on the network service of fog nodes, such as U2R on fog nodes. The ultimate goal of the intruder is to gain access to the cloud server and carry out further invasion. When an invader achieves

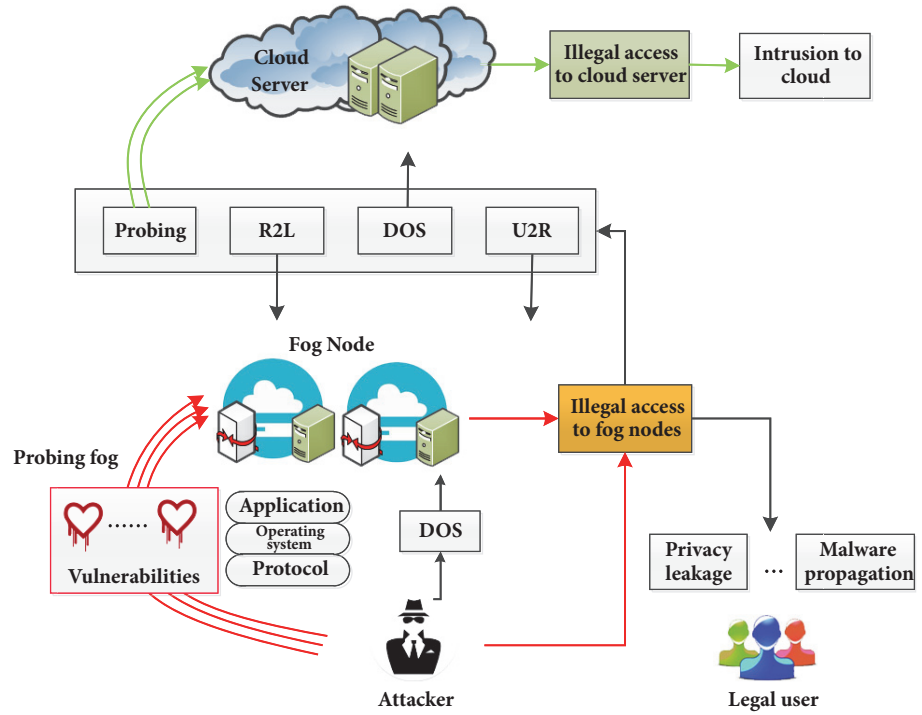


Figure 2: Intrusion process in fog computing.

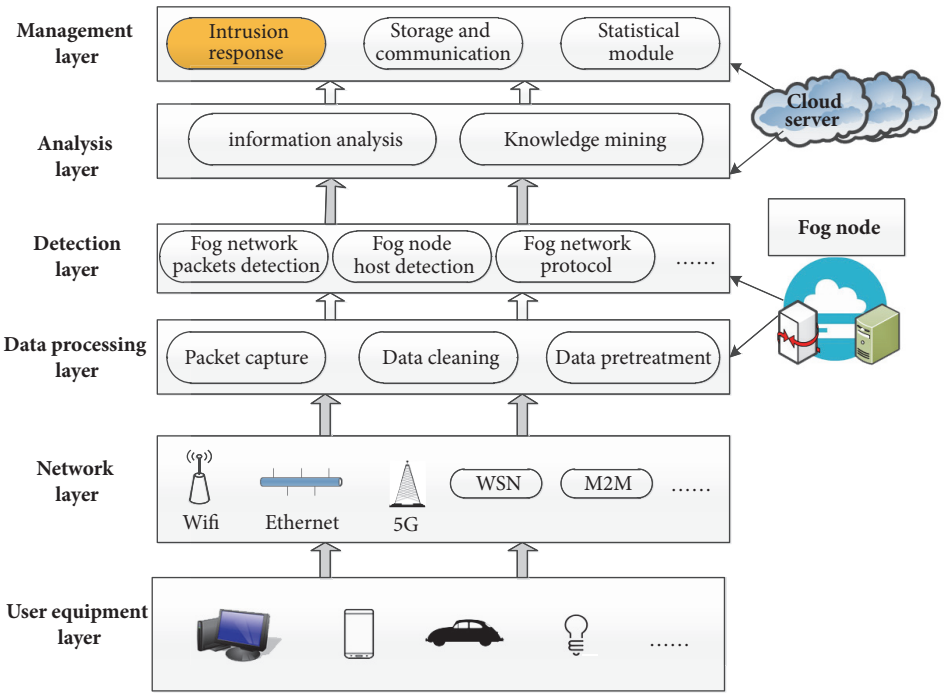


Figure 3: The general framework for fog computing intrusion detection system.

a user-to-root attack (U2R) on a fog node, the bugs of the cloud server will be continuously scanned and utilized by the invader to seek access to the cloud. Reducing the invasion frequency of invaders and improving the traffic of legal users in the system are against the invasion in fog computing. The focus of this study is on the intrusion response. Intrusion response is an important function of IDS framework [8] as shown in Figure 3.

The framework is a 6-layer IDS framework. It contains a series of functional modules, such as detection and response,

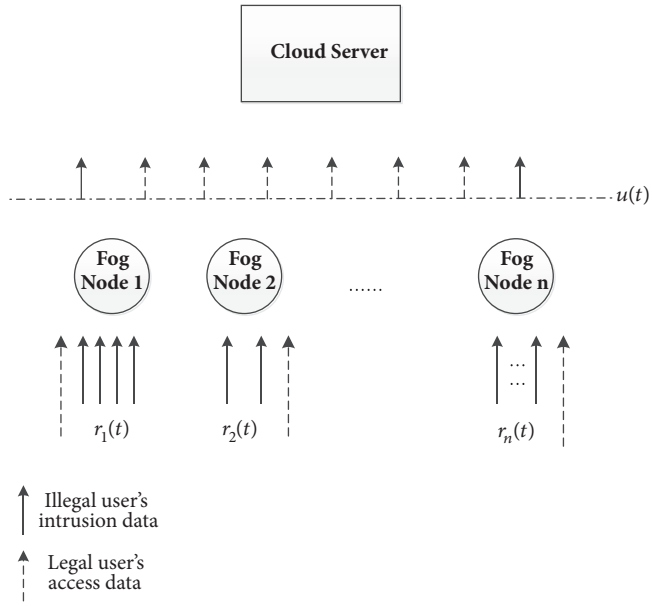


Figure 4: Attack strategy and response strategy in fog computing.

Table 1: The list of symbols' meanings.

Symbol	Notation
$r_i(t)$	Attacker's strategy: intrusion frequency for each fog node at time t, and it is the control variable of the attackers
$u(t)$	Defender's strategy: forbidding rate for system access at time t, and it is the control variable of the IDS
$x(t)$	System memory resource occupancy
W_A	The incomes of the attacker: expectation of invasion success
W_D	The incomes of the defender: access amount of legal users
J_A	The gains of the attacker
J_D	The gains of the defender
χ	Total access amount of system at t
α	Forbidding factor
θ	System capacity: $u(t)$ is enabled when the total access amount exceeds θ
β	Proportion of legal users in χ

which can ensure the security of fog computing. It shows that intrusion response is deployed in the cloud. The defense strategy of the cloud server is based on the whole system in order to defend against invasion from illegal users and minimize the loss of fog nodes after invasion. The invasion from illegal users and the response of the cloud server are regarded as the two players of attack-defense, and the problem is modeled and described in Section 3.

3. Differential Game Models

In this section, we model the two players of attack-defense in fog computing. In the process of invasion and response, intruders and system can be viewed as two players of the game, and their purpose is to maximize their benefits. The invader invades every fog node. The invasion frequency of each fog node is the attacker's strategy, aiming to access the fog node as much as possible so as to make illegal access. For defenders, restricting the invasion of illegal users and letting more legal users get access to fog network are the purpose of the system. The cloud server is mainly responsible for the implementation of the system response strategy. The forbidding rate of accessing users, u(t), is the control strategy of the cloud server. The process of intrusion and the process of response are shown in Figure 4.

The invaders and defenders in fog computing are analyzed and modeled, respectively, showing the relationship between the strategy of both invaders and defenders and their benefits. The list of symbols' meanings, which can be used during modeling, is shown in Table 1.

As shown in Figure 4, intruders start attack at fog nodes. The frequency of invasion against fog nodes is $r_i(t)$. The expectation of successful invasion is defined as the incomes of invaders, which is $W_A = r_i(t) \cdot \varphi_i(t)$, and $\varphi_i(t)$ is the probability of a single successful invasion. The probability of a round of successful invasion detected on the fog node i will increase when $r_i(t)$ is increased. When $r_i(t)$ increases, the probability of detecting attacks on the fog node will increase.

We define $\varphi(t) = 1 - r_i(t)/r_{i-\max}$, and $r_{i-\max}$ is the upper limit of the invasion frequency on the *ith* fog node. The incomes of invaders are

$$W_A = \sum_{i=1}^{n} r_i(t) \cdot \left[1 - \frac{r_i(t)}{r_{i-\max}}\right] \tag{1}$$

Cloud servers are as a defender of the system and the defense strategy is deployed from the perspective of fog cluster. The fog cluster is viewed as a system. When the number of system's access is too large, the system needs to take corresponding response strategy. The functional relationship between the system income and the access traffic of system is given and the functional relationship between the response strategy of system and the access traffic of system is also given:

$$W_D = \begin{cases} \beta\chi, & (\chi \le \theta) \\ [1 - u(t)] \cdot \chi, & (\chi > \theta) \end{cases} \tag{2}$$

θ is the capacity of the system. When $\chi \le \theta$, it represents the traffic of the legal access in the system. β represents the proportion of legal users in χ and it is obvious that $\beta \le 1$. $W_D = \beta\chi$. When $\chi > \theta$, it represents the traffic of the abnormal access in the system. Obviously, the proportion of illegal users is larger than legal access traffic. The original forbidding rate will no longer apply to the response of the system and the defense strategy $u(t)$ should be started at this time.

The larger χ is, the greater proportion of illegal users is. The forbidding peer rate of system needs to be larger. So $u(t) = \alpha \cdot \chi$, and α represents the forbidding factor.

When $\chi > \theta$

$$W_D = (1 - \alpha\chi) \cdot \chi = -\alpha\chi^2 + \chi \tag{3}$$

According to formula (3), W_D exhibits an impact of quadric relationship on χ. Based on the properties of quadratic function, we give Theorem 1.

Theorem 1. *When the system starts the defense policy, the maximum capacity of the system cannot exceed* $1/\alpha$.

Proof. When χ becomes large to $\chi = 1/\alpha$, $W_D = 0$. When χ continues to become larger, $W_D < 0$, the incomes of the system are negative and legal users cannot make access and the system is down. There is no meaning to start the strategy. □

Corollary 2. *Condition 1 for the system to take the response strategy is that the value of* χ *is between* θ *and* $1/\alpha$. $\chi \in (\theta, 1/\alpha)$.

The range of forbidding factor α is as follows:

Based on the properties of quadratic function, $\max(W_D) = 1/4\alpha$. $\theta \ge 1/4\alpha$ because W_D represents access amount of legal users. When $\chi = \theta$, the system reaches its critical point.

If $\theta < (1-\beta)/\alpha$, then $\beta\cdot\theta < -\alpha\cdot\theta^2 + \theta$. $\beta\cdot\theta$ is $\max\{W_D\}_{\chi\le\theta}$, which is the maximal incomes of system before taking the strategy.

From $(2\alpha\cdot\theta - 1)^2 \ge 0$, it can be concluded that $-\alpha\cdot\theta^2 + \theta \le 1/4\alpha$.

Because of the properties of quadratic function, $1/4\alpha$ is the maximal incomes of the system after taking the strategy.

$\beta \cdot \theta < 1/4\alpha$, which is $\max\{W_D\}_{\chi>\theta} \ge \max\{W_D\}_{\chi\le\theta}$.

The relationship between θ and α is as follows:

$$\frac{1}{4\alpha} \le \theta < \frac{1-\beta}{\alpha} \tag{4}$$

The inequality needs to satisfy $1/4\alpha < (1-\beta)/\alpha$, and the range of β is further determined: $\beta \in (0, 0.75)$

According to inequality (4), the range of forbidding factor α is as follows: $\alpha \in [1/4\theta, (1-\beta)/\theta)$.

Corollary 3. *Condition 2 for the system to take the response strategy* $u(t)$ *is as follows:* $\alpha \in [1/4\theta, (1-\beta)/\theta)$, $\beta \in (0, 0.75)$.

Corollary 4. *The maximal incomes of the system can be improved when the response strategy is used.*

Following the above, a precise function of income can be obtained:

$$W_D = \begin{cases} \beta\chi, & \chi \in [0, \theta] \\ [1 - u(t)] \cdot \chi, & \chi \in \left(\theta, \dfrac{1}{\alpha}\right] \end{cases} \qquad \alpha \in \left[\frac{1}{4\theta}, \frac{1-\beta}{\theta}\right), \ \beta \in (0, 0.75) \tag{5}$$

In the further discussion, when $\chi > \theta$, the relationship between incomes of the system and the strategy of the system is important. From $u(t) = \alpha \cdot \chi$, the income after the strategy being used is

$$W_D = -\frac{u^2(t)}{\alpha} + \frac{u(t)}{\alpha} \tag{6}$$

The process of game is dynamic and continuous. Both the invasion against fog nodes started by invaders and the corresponding strategy executed by the system have a direct impact on memory occupancy. Memory occupancy can be dynamically represented as $x(t)$:

$$\frac{dx(t)}{dt} = \left[\sum_{i=1}^{n} a r_i^2(t)\right] + b \cdot u(t) + x(t) \tag{7}$$

Because of $x(t)$, when the gains of both system and invaders are calculated, the impact on the gains caused by $x(t)$ needs to be considered. If the memory occupancy of system increases, this may be due to the resource occupation caused by the successful invasion. This is what invaders want, as the starter of the invasion from outside. The impact of $x(t)$ on the gains needs to be considered when the gains are calculated. On the contrary, for the system, the start of the response strategy requires system memory. $x(t)$ is as the cost of response when calculating gains, it should deduct the

impact of $x(t)$ on its gains from W_D. The object functions [16] of system gains and invader gains are

$$J_D = \max_{u(t)} \int_{t_0}^{T} e^{-r(t-t_0)} \left[(1-u(t)) \frac{u(t)}{\alpha} - c \cdot x(t) \right] dt + q_1 (x(T)) e^{-r(T-t_0)} \tag{8}$$

$$J_A = \max_{r_i(t)} \int_{t_0}^{T} e^{-r(t-t_0)} \left[\left(1 - \frac{r_i(t)}{r_{i-\max}} \right) r_i(t) + d \cdot x(t) \right] dt + q_2 (x(T)) e^{-r(T-t_0)} \tag{9}$$

c and d are influence factors of $x(t)$ on the gains of invaders and gains of the system.

4. Solution of Optimal Strategy

In Section 3, intruders attack and system defense in fog network are regarded as dynamic game processes, and a differential game model is established. In this section, the optimal strategy of differential game model established in Section 3 will be solved [16, 18].

From Bellman equation, formula (8) can be changed to

$$-V_t(t,x) = \max_{u(t)} \left\{ \left[(1-u(t)) \frac{u(t)}{\alpha} - c \cdot x(t) \right] e^{-r(t-t_0)} + V_x(t,x) \left[\left[\sum_{i=1}^{n} a r_i^2(t) \right] + b \cdot u(t) + x(t) \right] \right\} \tag{10}$$

$$V(T,x) = q_1 (x(T)) \exp\left(-r(T-t_0)\right) \tag{11}$$

From the derivation of formula (10) with respect to $u(t)$, we can obtain the optimal response strategy as

$$u^*(t) = \frac{1}{2} \left(1 + V_x(t,x) \alpha b e^{r(t-t_0)} \right) \tag{12}$$

We assume that

$$V(t,x) = e^{-r(t-t_0)} (A(t) x + B(t)) \tag{13}$$

Substituting formula (12) into (10) and (11), we can get

$$V_t(t,x) = e^{-r(t-t_0)} \left(-rA(t) + A'(t) \right) x + e^{-r(t-t_0)} \left(-rB(t) + B'(t) \right) \tag{14}$$

$$V_x(t,x) = e^{-r(t-t_0)} A(t) \tag{15}$$

Then the optimal intrusion response strategy can be represented in terms of $A(t)$ as

$$u^*(t) = \frac{1}{2} (1 + A(t) \alpha b) \tag{16}$$

where we need to solve the expression of $A(t)$ firstly.

$V_x(t,x)$ can be changed into

$$V_x(t,x) = \mathrm{e}^{-r(t-t_0)} \left(rA(t) x + rB(t) - A'(t) x - B'(t) \right) = \frac{1 + A(t) \alpha b}{2\alpha} - \frac{1}{4\alpha} (1 + A(t) \alpha b)^2 + A(t) \cdot \left[\sum_{i=1}^{n} \frac{a \left[r_i^*(t) \right]^2}{2} + \frac{b}{2} (1 + A(t) \alpha b) + x(t) \right] \tag{17}$$

from which we can deduce that

$$\begin{aligned} rA(t) - A'(t) &= A(t) - c \\ rB(t) - B'(t) &= \frac{1}{4\alpha} \left(1 - A^2(t) a^2 b^2 \right) + \frac{b}{2} (1 + A(t) ab)^2 + A(t) \sum_{\mathrm{i}=1}^{\mathrm{n}} a \left(\mathrm{r}_{\mathrm{i}}^*(\mathrm{t}) \right)^2 \end{aligned} \tag{18}$$

Formula (18) gives rise to the expression of $A(t)$

$$A(t) = q_1 e^{(r-1)(t-T)} + \frac{c \cdot e^{(r-1)(t-T)}}{r-1} - \frac{c}{r-1} \tag{19}$$

The optimal response strategy of system response is finally presented

$$u^*(t) = \frac{1}{2} (1 + \alpha b A(t)) = \frac{1}{2} + \frac{\alpha b}{2} \left[q_1 e^{(r-1)(t-T)} + \frac{c \cdot e^{(r-1)(t-T)}}{r-1} - \frac{c}{r-1}] \right] \tag{20}$$

Then we solve the optimal intrusion strategy. From Bellman equation, formula (9) can be changed into

$$-W_t(t,x) = \max_{r_i(t)} \left\{ \left[r_i(t) \left(1 - \frac{r_i(t)}{r_{i-\max}} \right) + d \cdot x(t) \right] e^{-r(t-t_0)} + W_x(t,x) \left[\left[\sum_{i=1}^{n} a r_i^2(t) \right] + b \cdot u(t) + x(t) \right] \right\} \tag{21}$$

$$W(T,x) = q_2 (x(T)) \exp\left(-r(T-t_0)\right) \tag{22}$$

In formula (21), the derivation with respect to $r_i(t)$ leads to the optimal intrusion strategy

$$r_i^*(t) = \frac{r_{i-\max}}{2 - 2 r_{i-\max} C(t) a} \tag{23}$$

In the same way, to solve the expression of $C(t)$, we assume

$$W(t,x) = e^{-r(t-t_0)} (C(t) x + D(t)) \tag{24}$$

Table 2: Range of parameters in the model.

Parameters		θ	β	α	a	b	c	d	r	q_1	q_2	$r_{i-\max}$
Range of value	Upper limit	250	0.5	0.001	0.5	20	0.1	0.01	0.1	0.2	0.8	50
	Lower limit			0.002	5	100	0.5	0.1	0.9			100

Substituting formula (23) into (21) and (22), we can get

$$W_t(t,x) = e^{-r(t-t_0)}\left(-rC(t)x + C'(t)x - rD(t) + D'(t)\right) \tag{25}$$

$$W_x(t,x) = e^{-r(t-t_0)}C(t) \tag{26}$$

From formula (24), $W(t,x)$ can be written as

$$\begin{aligned} W(t,x) = {} & \frac{r_{i-\max}}{2 - 2r_{i-\max}C(t)a} \\ & - \left[\frac{r_{i-\max}}{4\left(1 - r_{i-\max}C(t)a\right)^2}\right] + C(t) \\ & \cdot \left[\sum_{i=1}^{n} \frac{ar^2{}_{i-\max}}{4\left(1 - r_{i-\max}C(t)a\right)^2} + bu^*(t)) + x(t)\right] \\ & - d \cdot x(t) \end{aligned} \tag{27}$$

From formula (27), we can deduce that

$$\begin{aligned} rC(t) - C'(t) &= d + C(t) \\ rD(t) - D'(t) &= Q \end{aligned} \tag{28}$$

where $Q = r_{i-\max}/(2 - 2r_{i-\max}C(t)a) - r_{i-\max}/4(1 - r_{i-\max}C(t)a)^2 + \sum_{i=1}^{n}(C(t)ar^2{}_{i-\max}/4(1 - r_{i-\max}C(t)a)^2) + (bC(t)/2)(1 + A(t)\alpha b)$.

We can get the expression of $C(t)$ as

$$C(t) = q_2 e^{(r-1)(t-T)} + \frac{d \cdot e^{(r-1)(t-T)}}{r-1} - \frac{d}{r-1} \tag{29}$$

Therefore, the optimal invasion strategy of each fog node is

$$r_i{}^*(t) = \frac{r_{i-\max}}{2 - 2r_{i-\max}\left[q_2 e^{(r-1)(t-T)} + d \cdot e^{(r-1)(t-T)}/(r-1) - d/(r-1)\right]a} \tag{30}$$

From formula (7), we can update the equation of state

$$\begin{aligned} \frac{dx^*(t)}{dt} = {} & \left[\sum_{i=1}^{n} a\left(\frac{r_{i-\max}}{2 - 2r_{i-\max}\left[q_2 e^{(r-1)(t-T)} + d \cdot e^{(r-1)(t-T)}/(r-1) - d/(r-1)\right]a}\right)^2\right] \\ & + b\left[\frac{1}{2} + \frac{\alpha b}{2}\left[q_1 e^{(r-1)(t-T)} + \frac{c \cdot e^{(r-1)(t-T)}}{r-1} - \frac{c}{r-1}\right]\right] + x^*(t) \end{aligned} \tag{31}$$

In summary, we get the optimal response strategy and the optimal intrusion strategy. They are described with formula (20) and (30), respectively.

5. Numerical Simulation

In this section, Matlab 2014a software is used for simulation. According to the model, the tendency of the optimal response strategy changing with time and the tendency of the optimal invasion strategy changing with time are analyzed, respectively. In the limited time domain, 5 fog nodes are analyzed in 20 minutes to obtain the dynamic rules of system strategy and intruder strategy. The parameters used in the simulation are shown in Table 2.

The access capacity of the system is $\theta = 250$ and the proportion of legal users in the system is $\beta = 0.5$. From (5), the range of α is fixed. $r_{i-\max}$ is the invasion limit on each fog node, its domain is between 50 and 100. For a and b, they represent the influence factors of intrusion strategy and the influence factors of response strategy on $x(t)$, respectively. Since the influence of intruders on $x(t)$ is indirect and the response strategy adopted by the system is direct to $x(t)$, assuming a is less than b, d and c are the influence factor of $x(t)$ on gains of invaders and the influence factor of $x(t)$ on gains of system so c needs to be between 0.1 and 0.5 and d needs to be between 0.01 and 0.1. r is a discount factor, which needs to be between 0 and 1.

First, the response strategy of the system $u(t)$ is stimulated. Figure 5 shows the changing rule of $u(t)$ in different limiting factors α. It can be seen that the value of α has an impact on the initial value of the game. However, when the game begins, no matter what α is, $u(t)$ will converge and be stable at around 10 minutes. This shows that when the system capacity of θ and β is fixed, the value of α has a little influence on the option of response strategy, which means when choosing strategy, the attempts to choose limiting factor α do not have to be frequent, for the cloud server choosing the strategy.

Figure 6 shows the convergence of the optimal defense strategy of the system when the discount factor r takes

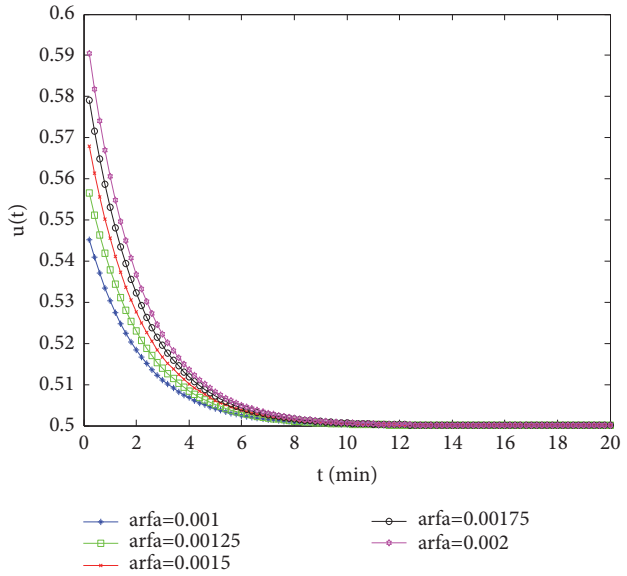


Figure 5: The variation of optimal response strategy over time with different α.

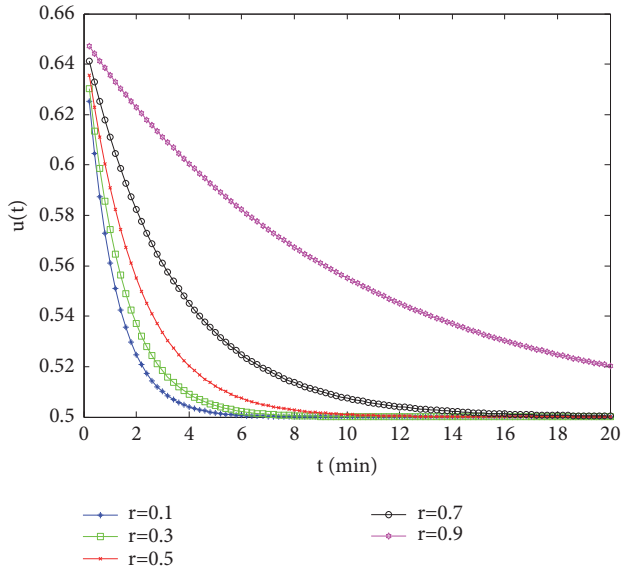


Figure 6: The variation of optimal response strategy over time with different r.

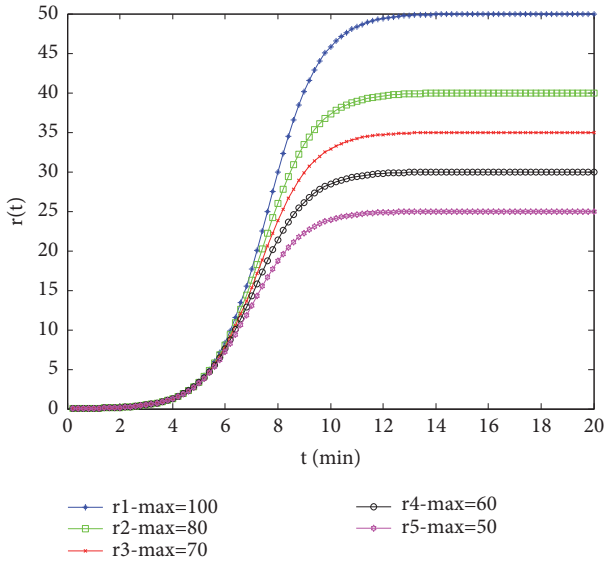


Figure 7: The optimal intrusion strategy when 5 fog nodes are invaded.

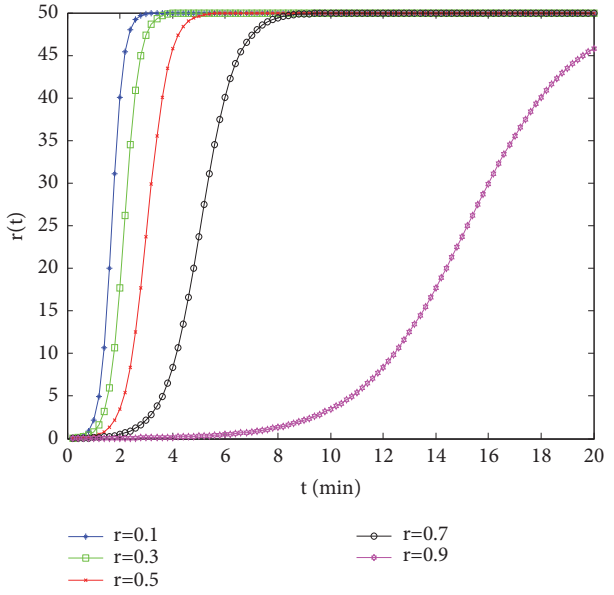


Figure 8: The variation of optimal intrusion strategy over time with different r.

different values. The smaller r is, the faster optimal strategy converges. When r is 0.9, the terminal time of the game is still not convergent. The reason for it is that r, as the main parameter of the ultimate gains, will directly affect choosing system response strategy in the process of dynamic game.

The optimal intrusion strategy of the invaders is stimulated, assuming that an invader invades 5 fog nodes and Figure 7 shows five intrusion strategies against 5 fog nodes of different $r_{i-\max}$. At the initial time, the frequency of invasion is 0. As time goes on, the invaders will increase the frequency of invasion to gain higher gains. However, as the game continues, the invasion strategy $r_i(t)$ will also reach a steady state. Similar to the system response strategy, it also converges at a certain time. Similarly, in order to observe the convergence of $r_i(t)$, discount factor r takes different values. Figure 8 shows the tendency of optimal intrusion strategy changing with time when r takes different value. Obviously, the smaller r is, the faster the optimal strategy converges.

A comprehensive analysis shows that the system broadens the restrictions on access traffic in the process of reducing the forbidding rate $u(t)$ of the system. At the same time, in order to maximize the incomes, the intruders will also enhance the intrusion level on the fog nodes. As the game continues, the system and intruders will adjust their strategy to maximize their incomes. The system state $x(t)$ will also change when

the strategy is changed. $r_i(t)$ and $u(t)$ will also affect the incomes of players. Therefore, this process is a game between two players adjusting the optimal strategy and making the optimal strategy converge.

6. Conclusions

Fog computing is a new computing paradigm, and its security problem can not be ignored. As the manager of fog cluster, cloud server needs to respond in time when intrusion occurs. Firstly, the characteristics of intrusion in fog computing are analyzed, and the invaders and system in fog computing are modeled, respectively. Then the differential game model is solved, and the optimal strategy of intruders and system is obtained. Finally, we simulated the optimal intrusion strategy and the optimal response strategy, and we analyzed the experimental results. The results show that our game model and the optimal strategy can guarantee the security of fog cluster.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Key R&D Program of China (2017YFC0820700), the Foundation of Science and Technology on Information Assurance Laboratory (no. KJ-17-101), and the National Science Foundation Project of China (no. 61701020).

References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop, MCC 2012*, pp. 13–16, ACM, Helsinki, Finland, August 2012.

[2] Y. Huo, C. Yong, and Y. Lu, "Re-ADP: Real-time Data Aggregation with Adaptive w-event Differential Privacy for Fog Computing," *Wireless Communications and Mobile Computing*, pp. 1–13, 2018.

[3] J. Su, F. Lin, X. Zhou, and X. Lu, "Steiner tree based optimal resource caching scheme in fog computing," *China Communications*, vol. 12, no. 8, Article ID 7224698, pp. 161–168, 2015.

[4] Y. Huo, Y. Tian, L. Ma, X. Cheng, and T. Jing, "Jamming Strategies for Physical Layer Security," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 148–153, 2018.

[5] B. Z. Abbasi and M. A. Shah, "Fog computing: Security issues, solutions and robust practices," in *Proceedings of the 2017 23rd International Conference on Automation and Computing (ICAC)*, pp. 1–6, IEEE, Huddersfield, UK, 2017.

[6] D. E. Denning, "An Intrusion-Detection Model," in *IEEE Symposium on Security and Privacy*, IEEE, Oakland, CA, USA, 1986.

[7] Y. Huo, C. Hu, X. Qi, and T. Jing, "LoDPD: A Location Difference-Based Proximity Detection Protocol for Fog Computing," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1117–1124, 2017.

[8] X. An et al., "Sample Selected Extreme Learning Machine Based Intrusion Detection in Fog Computing and MEC," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7472095, 10 pages, 2018.

[9] Lin. Fuhong et al., "Fair Resource Allocation in Intrusion Detection System for Edge Computing," *IEEE Consumer Electronics Magazine*, 2018.

[10] N. Stakhanova, S. Basu, and J. Wong, "A taxonomy of intrusion response systems," *International Journal of Information and Computer Security*, vol. 1, no. 1-2, pp. 169–184, 2007.

[11] Y. B. Guo and M. A. J. Feng, "Game theoretical framework for adaptive intrusion detection and response," *System Engineering and Electronics*, 2005.

[12] S. A. Zonouz et al., "RRE: A Game-Theoretic Intrusion Response and Recovery Engine," *IEEE Transactions on Parallel and Distributed System*, vol. 25, no. 2, pp. 395–406, 2013.

[13] S. Jin, L. Yin, and A. X. Li, "Dynamic Intrusion Response Based on Game Theory," *Journal of Computer Research and Development*, vol. 45, no. 5, pp. 747–757, 2013.

[14] W. P. Wang and W. W. Zhu, "Network Security Behavior Model Based on Dynamic Non-Cooperative Game Model with Incomplete Information," *Journal of Chinese Computer System*, vol. 27, no. 2, pp. 253–256, 2006.

[15] A. Kundu and S. K. Ghosh, "Game Theoretic Attack Response Framework for Enterprise Networks," in *Distributed Computing and Internet Technology*, vol. 8337, pp. 263–274, Springer International Publishing, New york, NY, USA, 2014.

[16] D. W. Yeung and L. A. Petrosyan, *Cooperative Stochastic Differential Games*, Springer, New York, NY, USA, 2006.

[17] Z. Li, X. Zhou, Y. Liu, H. Xu, and L. Miao, "A non-cooperative differential game-based security model in fog computing," *China Communications*, vol. 14, no. 1, pp. 180–189, 2017.

[18] A. Dixit, "A model of duopoly suggesting a theory of entry barriers," *Bell Journal of Economics*, vol. 10, no. 1, pp. 20–32, 1979.