IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 August 2024, accepted 28 August 2024, date of publication 5 September 2024, date of current version 16 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3454823

RESEARCH ARTICLE

On the Security of Distributed Multi-Agent *K*-Means Clustering With Local Differential Privacy

CONGCONG SHI, XIULI HUANG, AND PENGFEI YU
State Grid Laboratory of Power Cyber-Security Protection and Monitoring Technology, State Grid Smart Grid Research Institute Company Ltd., Nanjing 210003, China

Corresponding author: Xiuli Huang (geiria307@163.com)

This work was supported by the Science and Technology Project of State Grid Corporation of China under Grant 5108-202218280A-2-393-XG.

ABSTRACT In a distributed scenario, the process of multiple agents collaborating and interacting with the server to iteratively implement k-means clustering analysis can be easily exploited by attackers, posing a huge privacy threat. Therefore, a local differential privacy k-means method (LDPKmeans) was proposed, which can effectively address the privacy protection problem in multi-agent systems. In this paper, we propose an effective attack method based on multi-agent model, which shows that the basic proposal of LDPKmeans will leak the real information of user agents if the attacker only obtains the cluster information and cluster centroid of each user. Furthermore, we enhance the attack method to crack the improved LDPKmeans method with privacy enhancement, enabling us to infer the cluster information of each user agent in the server. In other words, LDPKmeans seriously leak user agent privacy in distributed multi-agent systems if the server is untrusted. Theoretical analysis and experiments evaluate the effectiveness of our attack scheme. The results show that our method can effectively attack the distributed LDPKmeans scheme compared with the state-of-the-art attack methods. Specifically, our attack method can reduce the average relative error of inferring the true value before k-menas convergence on the 3D Road Network and Shuttle datasets by about 54% and 75% respectively when $k = 5$.

INDEX TERMS Multi-agent systems, distributed k-means, local differential privacy, security problem.

I. INTRODUCTION

In multi-agent systems [1], [2], [3], communication and learning among various autonomous agents [4] involve privacy and security issues when each agent can autonomously choose which information to share and collaborate together to achieve the overall goal. In a system scenario with multiple user agents and one server, the application of k-means clustering [5], [6], [7] operations can easily cause the data saved during the operation to be exploited by untrusted third parties, posing a huge privacy threat. It is crucial for multi-agent systems [1], [8] to preserve privacy while participating in clustering, as the data of each agent may contain sensitive private information. To protect user privacy, privacy-preserving implementations of iterative clustering algorithms have been extensively studied.

There are numerous studies [6], [9], [10], [11], [12], [13], [14], [15] in the field of privacy-preserving multi-agent clustering that leverage public-key encryption algorithms [16]. The work [9] uses the results of secure multiparty computation for vertically split entity data to generate a privacy-preserving k-means clustering without disclosing any values on which the clustering is based. A secure two-party k-means clustering protocol with guaranteed privacy is proposed in [10], which can efficiently compute information for multiple iterations of k-means clustering without revealing intermediate values. Although privacy protection methods using public-key encryption [17] for clustering can provide strong security guarantees to the data, they may lead to problems such as clustering inaccuracy, high computational cost and data unavailability.

The associate editor coordinating the review of this manuscript and approving it for publication was Miguel López-Benítez.

To enable multiple user agents to provide sufficient data utility without leaking sensitive information of the agent itself, differential privacy (DP) [18], [19], [20], [21], [22] is widely used to achieve formal privacy-preservation in distributed multi-agent clustering algorithms [5], [19], [23], [24]. Su et al. [19] propose a non-interactive DP clustering algorithm (EUGkm) and an improved version of the interactive DP k-means algorithm, which improved the selection of the initial point and limited iteration of the k-means algorithm times. Jones et al. [25] consider the DP variants of k-means and k-median in the metric space, and further improved the applicability of the DP clustering algorithm, making it easier to incorporate DP into existing clustering in the multi-agent frame. However, the aforementioned privacy-preserving clustering algorithms are based on the assumption of trusted servers, which may not be applicable in the real world where third-party servers cannot be fully trusted.

Xia et al. [5] first proposed the k-means clustering method of local differential privacy (LDP) [26], [27], [28], [29], [30] in a distributed multi-agent system, referred to as LDPKmeans. LDPKmeans allows user agents and servers to interactively perform private k-means clustering. Specifically, the user agent is primarily responsible for perturbing the data and computing the clusters to which the data belongs. Afterward, the user agents submit this perturbed information to the server for further processing. Then, the server aggregates the perturbed data, performs clustering, updates the cluster centroids, and sends the updated cluster centroids back to the user agent after each round of updates. The two parties continue to interact until the algorithm converges, ensuring that individual user privacy is not compromised while maintaining the accuracy of k-means clustering results. After that, Sun et al. [31] study the problem of non-interactive clustering in a distributed environment under a LDP framework. They encoded the user agent source data and embedded it into the anonymous Hamming space, then added noise to provide the indistinguishability of LDP, and used the distance-aware estimation algorithm on the untrusted server side to directly integrate and cluster the perturbed data. The difference between [5] and [31] lies in the fact that in [31], the server agent does not require multiple rounds of interaction with the user agent. Instead, it only needs to implement clustering based on the distance estimation algorithm, which reduces the communication cost significantly. Under this distributed privacy protection k-means framework, it is actually a collaborative operation of a multi-intelligent system, that is, multiple users and servers are regarded as interactions between multiple agents. Each agent is autonomous, it can decide which data to send to another agent. Reference [2] solves the problem of maximizing prediction accuracy while ensuring privacy in two-agent systems in a decentralized manner using distributed learning or federated learning. Reference [32] designed a new k-means algorithm running under LDP, which can significantly reduce the additive error and maintain the multiplicative error produced by k-means operation.

Due to our keen interest in the distributed multi-agent k-means algorithm under the LDP framework, we conduct a detailed analysis of [5]. Our findings revealed that the k-means clustering in [5] requires multiple iterations to achieve convergence, and there are several interactions between the client agent and the service provider. In each iteration, the user agent is required to locally recalculate the cluster information to which the real data belongs and submit it to the service provider. Despite not submitting real data values, this cluster information is highly correlated with the actual data. It could be easily learned and exploited by attackers, potentially leading to the leakage of private information and causing significant losses to property and reputation. When multiple user agents send disturbance information to the server, a malicious server can monitor or record these communication information. For example, assuming that what the user submits is location data, the user will submit the scope of the real data for each round of iteration. After multiple rounds of iterations, it is easy for an attacker to infer the real data by pinpointing the real location of the user to a small range. When multiple user agents submit real cluster information to the server multiple times, the malicious agent will store the information of each round. Combined with the stored centroid information for each iteration, they infer the sensitive information of a specific user agent.

In this paper, we investigate the privacy and security of LDPKmeans, and demonstrate that LDPKmeans can be compromised upon each round of centroid data and cluster information recorded by the server. Firstly, we demonstrate that LDPKmeans leaks user agents privacy over multiple iterations if the attacker gains access to the cluster information submitted by the user and the cluster centroids of each round. Secondly, we propose two attack strategies against the basic proposal of LDPKmeans, considering whether the server agent is malicious or not. Finally, we identify the security vulnerabilities of improved LDPKmeans with privacy enhancement, providing evidence that the enhancement scheme still results in the leakage of user agents privacy. Generally, our contributions in this paper can be summarized as follows:

1) We propose a novel attack method based on multi-agent model to break distributed LDPKmeans while the attacker obtains the clustering information of the data points and the cluster centroids of each iteration. Based on our attack, we can accurately find data points that are highly similar to the real value in the distributed privacy-preserving multi-agent k-means clustering.
2) Furthermore, we introduce an enhanced inference attack method to crack the improved LDPKmeans with privacy enhancements mentioned in [5]. This attack method is capable of recovering the true cluster information of each user, and gaining highly-accurate real value.

3) We evaluate the proposed attack scheme through theoretical and extensive simulations, and the results show that our proposed attack method can quickly crack LDPKmeans and its enhanced version, and infer the true value with an average accuracy higher than 50% compared with the state-of-the-art.

The rest of the paper is organized as follows. Section II introduces the system model, basic definition, and LDP-Kmeans. Section III proposes the detail of our attack approach based on multi-agent model. Then, Section IV describes the experiment evaluation of our attack schemes and response strategy. Section V reviews the related work. Finally, Section VI concludes this paper.

II. PROBLEM STATEMENT AND PRELIMINARIES

A. PROBLEM STATEMENT

In this paper, we consider a system involving two independent parties: several local agents (user) and a service provider (server). Fig. 1 shows that in multiple rounds of distributed k-means, the server will save the information of each round of iteration, which can be easily used by attackers to infer the true value.

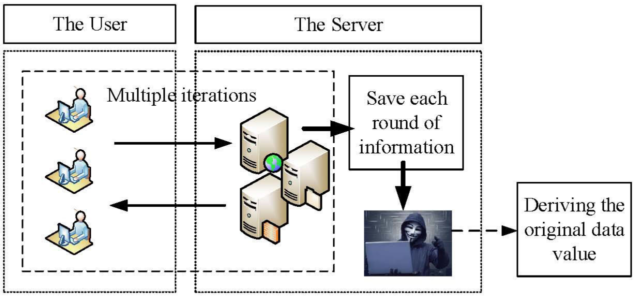

The User
The Server
Multiple iterations
Save each round of information
Deriving the original data value

FIGURE 1. The information of each round of the distributed k-means scheme can be easily exploited by attackers.

Table 1 gives the frequently used symbols and their descriptions. Given user agent set $U = \{u_1, u_1, \cdots, u_n\}$, $V = \{v_1, v_2, \cdots, v_n\}$, represents the user's data feature vector. Among them, each data vector u_i corresponds to m features $v_i = (x_{i1}, x_{i2}, \cdots, x_{im})$, $m = |v_i|, x_{ij} \in R, 1 \leq j \leq m$. To deal with untrustworthy service providers and protect the privacy of individual data, users want to perturb the individual data locally, and then submit the perturbed data vector $\tilde{V} = \{\tilde{v}_1, \tilde{v}_2, \cdots, \tilde{v}_n\}, \tilde{v}_i = (\tilde{x}_{i1}, \tilde{x}_{i2}, \cdots, \tilde{x}_{in})$ to the service provider. The service provider aggregates all perturbed data and performs k-means clustering, attempts to divide all data into K groups $\mathbb{C} = \{\mathbb{C}_1, \mathbb{C}_2, \cdots, \mathbb{C}_K\}$, and publishes K clusters to all users in each iteration class centroid. Let $C = \{C_1, C_2, \cdots, C_K\}$ be the cluster centroid, $C_k = (c_{k1}, c_{k2}, \cdots, c_{km})$, $k = 1$ to K. Each user finds the nearest centroid C_k through local calculation, and then returns the current cluster index (such as: the kth cluster) as the information of the cluster it belongs to, and sends it to the service provider. The service provider updates the cluster centroid according to the perturbed data and the cluster information until the clustering result converges.

TABLE 1. List of frequently used notations.

Notation	Description
U	User agents
V	User data collection
v_i	A user data vector with multiple features $v_i = (x_{i1}, x_{i2}, \cdots, x_{im})$
$\tilde{v}_i$	A perturbed user data vector
$\tilde{V}$	Perturbed user data
n	Number of user data
m	Number of data features
ε	Privacy budget
k	Cluster size
$\mathbb{C}$	Cluster
C	The collection of cluster centroids, $C = \{C_1, C_2, ..., C_k\}$
C_k	Centroid vector value $C_k = (c_{k1}, c_{k2}, \cdots, c_{km})$
$\mathcal{F}$	Random mechanism
P	Probability
B_{ij}, s	Binary String
T	Iterations
R_t	The radius of the tth iteration

The security goal requires that the privacy of raw user data should not be violated in the above-mentioned distributed k-means clustering analysis, that is, no useful information about any u_i should be exposed. On the contrary, the attacker's goal is to use the obtained information to infer the user's real data as accurately as possible. Malicious service providers will record the cluster information submitted by each user and the cluster centroid of each round of iteration, and then use the distance relationship between data points and cluster centroids to achieve the goal of reasoning attacks on real data. In this paper, we consider "attacker", "malicious server agent" and "service provider" to be the same and their meanings are interchangeable. The detailed description of the adversary will be provided in Section III.

B. BASIC DEFINITION

LDP inherits the basic definition and properties of centralized DP definition, and can eliminate the dependence on trusted servers while protecting user privacy in the process of data collection. Below we give a specific definition of LDP.

Definition 1 (ε-LDP): A random mechanism $\mathcal{F}(\cdot)$ satisfies ε-LDP($\varepsilon > 0$), if and only if any two user data $v, v' \in D$, we have

$$\forall s \in \mathcal{F}(D) : Pr[\mathcal{F}(v) = s] \leq e^{\epsilon} \cdot Pr[\mathcal{F}(v') = s] \qquad (1)$$

Among them, $\varepsilon \geq 0$ is the privacy budget, which is used to control the degree of privacy protection. $\varepsilon - LDP$ is a local setting of DP and does not depend on a trusted data curator.

Random Response: The main idea of LDP is that the user perturbs the original data locally through a random mechanism. Random response technology is the mainstream method to realize LDP, which only responds to discrete data containing two values. We introduce random response using a questionnaire with sensitive questions. Suppose the investigator wants to know how many users have AIDS. He will directly ask "Are you an AIDS patient?". Each user will flip an uneven coin before answering the question. If the coin comes up heads, the user will tell the poller the true answer. Otherwise, users will provide the opposite answer to the investigator. This has been shown to satisfy ε-DP when the user flips a coin with probability P of heads:

$$P = \frac{e^{\varepsilon}}{1 + e^{\varepsilon}} \tag{2}$$

C. LDPKmeans MECHANISM

The distributed LDP k-means algorithm is completed by the interaction and cooperation between the user agent and the server agent, and no longer assumes the existence of a trusted third-party server. The steps of LDPKmeans [5] are as follows.

User Agent Operations: The user agents mainly completes perturbation of user data and calculate cluster information.

1) Feature transformation. In the LDPKmeans scheme, the processed data feature values are floating point numbers. They first need to convert decimal floating-point numbers to binary numbers, which involves the precision of floating-point numbers. Thus, they set a coefficient λ to scale and truncate floating point values to integers, where λ also determines the length of the binary string. Specifically, they convert the feature x_{ij} of user u_i into a binary string $B_{ij} = (b_1, b_2, \ldots, b_{l_j})$, $l_j = |B_{ij}|$ is the length of the binary string B_{ij}. The conversion formula is as follows:
$$x_{ij} = \lambda \cdot (2^{l_j} b_1 + 2^{l_j-1} b_2 + \cdots + 2^1 b_{l_j}) \tag{3}$$
2) Data perturbation. A random response mechanism is performed on the user vector in binary form, and each bit is randomly flipped according to the following probability. The probability of flipping is given below:
$$\widetilde{b} = \begin{cases} 1, & Pr = \frac{1}{2f} \\ 0, & Pr = \frac{1}{2f} \\ b, & Pr = 1 - f \end{cases}, \tag{4}$$
where $f = \frac{2}{e^{\varepsilon}+1}$ is the corresponding privacy parameter, which determines the probability of random flipping. After all bits are perturbed, the user calculates the perturbed decimal data value $\widetilde{v}_i = (\widetilde{x}_{i1}, \widetilde{x}_{i2}, \cdots, \widetilde{x}_{im})$ according to $\widetilde{x}_{ij} = \lambda \cdot (2^{l_j}\widetilde{b}_1 + 2^{l_j-1}\widetilde{b}_2 + \cdots + 2^1\widetilde{b}_{l_j})$. Finally, the user submits it to the service provider.
3) Calculation of cluster information. When the algorithm iterates once, the service provider will send the selected K cluster centroids to the local user. Each user calculates the distance to each centroid based on real data, and finds the cluster index closest to it: $argmin_k(\|\widetilde{v}_i - C_k\|), k = 1, \cdots, K$. At this point, the cluster information to which each user belongs can be obtained and sent to the service provider.

Server Agent Operations: The server side completes the selection of the initial cluster centroid, the grouping of the perturbed data and the update of the cluster centroid.

1) Selection of initial cluster centroids. After receiving the data, the service provider selects a set of centroids to send to the user according to the initialization of the data domain.
2) Group perturbed data. The service provider groups the perturbed data according to the cluster information to which the user data belongs. Then the perturbation data is converted into binary form according to the $\widetilde{B}_{ij}$ coefficient λ.
3) Update cluster centroids. For each $\mathbb{C}(k = 1, \cdots, K)$, the cluster centroids is $C_k = (c_{k1}, c_{k2}, \ldots, c_{km})$, the cluster size is $num_k = |C_k|$, the service provider counts the sum of each bit $\widetilde{sum}_{ir}^k = \sum_{\widetilde{v}_i \in \mathbb{C}_k} \widetilde{B}_{ij}[r], i = 1, \cdots, n; r = 1, \cdots, l_j$. According to the perturbation rule, $\widetilde{sum}_{ir}^k$ consists of two parts:
$$\widetilde{sum}_{ir}^k = sum_{ir}^k \cdot (1 - f) + num_k \cdot \frac{1}{2} f \tag{5}$$
The service provider then estimates the true value based on the statistical value:
$$sum_{ir}^k = \frac{\widetilde{sum}_{ir}^k - (num_k \cdot \frac{1}{2} f)}{1 - f} \tag{6}$$
Next, each centroid component can be updated as $c_{kj} = \lambda \cdot \sum_{r \in [1, l_j]} (2^{l_j - r \cdot \frac{sum_{ir}^k}{num_k}}), j \in [1, m]$. When num_k is larger, the estimated centroid component is more accurate. After updating the cluster centroids of each cluster, the service provider releases a new round of centroids to local users.

Together, users and service providers collaboratively perform the clustering process until the cluster centroids in two consecutive iterations become stable, indicating convergence of the clustering algorithm.

For the security, the basic proposal of [5] provided the cluster information of the real data in each iteration, and the untrusted server can easily record the centroid and cluster information of each round to infer the real data, thereby inferring the sensitive information of the user. Even though [5] tried to propose a privacy enhancement scheme claiming to solve the problem of information leakage of clusters, but after analysis, it was found that there are still security risks. In the enhanced scheme of LDPKmeans, even though the user does not directly submit the real cluster information to the server, the perturbed data and indicator vectors submitted during each iteration can still implicitly reveal user information. As a result, the attacker can infer the cluster information of the real data by analyzing the density of different position segments of the submitted perturbed data. Indeed, the LDPKmeans scheme exhibits serious privacy vulnerabilities, resulting in the leakage of user privacy.

III. OUR PROPOSED ATTACK SCHEMES TO LDPKmeans

A. ATTACKS TO THE BASIC LDPKMEANS

We consider the attack on the Basic Proposal of the LDPKmeans [5]. Intrinsically, we assume that there are two attack states for server. On the one hand, the server might be honest but curious, truthfully recording all relevant data information submitted by each user. This information can be easily exploited by third-party attackers. On the other hand, the server might be malicious, possessing a high degree of knowledge about user background information, and using this knowledge to efficiently infer user information. For these two cases, we propose different attack strategies respectively. The attack goal of these methods is to infer the user's true value v_i as accurately as possible during the iteration process of the LDPKmeans basic scheme.

1) ATTACKS BY HONEST-BUT-CURIOUS SERVICE PROVIDERS

In basic proposal of LDPKmeans, the server send K initial cluster centroids (the current iteration number is 1) $C^{(1)} = \{C_1, C_2, \cdots, C_K\}$ to all users on the local side. Each user calculates the distance between the raw data $v_i = (x_{i1}, x_{i2}, \cdots, x_{im}), i \in \{1, \cdots, n\}$ and the centroid $C_k = (c_{k1}, c_{k2}, \cdots, c_{km}), k \in \{1, \cdots, K\}$ on the local side, and returns the cluster index with the smallest distance, which is the cluster to which the current user data belongs.

Subsequently, all users on the local side send all cluster indexes to the server, and the server will faithfully record the information submitted by users during each round of iteration. Assuming that the cluster to which v_i belongs is the $1st$ cluster, we give an attack example of applying LDPKmeans basic proposal to user data. As shown in Fig. 2, we show the basic process of $2D$ data points inferring the original data based on cluster information and centroid values on the planar graph. Specifically, it is assumed that the basic scheme of LDPKmeans requires T rounds of iterations. Also, we draw a circle with radius R taking the distance from a certain cluster centroid to the farthest user. When $T = 1$, the attacker knows that v_i belongs to the $1st$ cluster $\mathbb{C}_1^{(1)}$. We take the center of mass $C_1^{(1)}$ as the center, and draw a circle with the distance from $C_1^{(1)}$ to the farthest user point in the cluster as the radius (subsequent iterations form a circle in the same way). When $T = 2$, the cluster to which v_i belongs is still the $1st$ cluster $C_1^{(2)}$, the cluster centroid is denoted as $C_1^{(2)}$. At this time, two consecutive rounds of iterations lead to an intersecting region between clusters $C_1^{(1)}$ and $C_1^{(2)}$ that contains v_i. We determine the center point O_1 of the intersection area through the intersection point between the line connecting the two centroids and the line connecting the intersection points of the two circles. Then we take O_1 as the center and the distance from O_1 to the intersection of the two circular areas as the radius to find a new circular area. The circular area here greatly reduces the range to which the actual value v_i belongs.

Similarly, when $T = 3$, we can shrink v_i to a smaller circular area with O_2 as the center. In summary, Fig. 2 shows how to reduce the specific range of the true value in each round of iteration. As the number of iterations T increases, the k-means clustering gradually converges, and at the same time, the feasible solution in the small range gradually approaches the original value.

Additionally, according to the basic characteristics of k-means clustering, when the number of iterations reaches a certain value, the change of the centroid is actually very small. Therefore, our attack method can construct a distance optimization equation to estimate the true value of the original user according to the centroid data of multiple iterations and the existing cluster information before k-means converges.

From the above example of two-dimensional data points, the true value v_i can be limited to a very small area through the centroid of multiple rounds of iterations and known cluster information. Finally, we find the point $\bar{v}_i$ closest to each round of centroid $C_i^{(t)}$ in this area, and $\bar{v}_i$ is the estimated value of the real data. Here, we generalize $2D$ planar points to multidimensional data points to make it more practical. We take the point $\bar{v}_i$ with the shortest distance to the cluster centroid $C_k^{(i)}$ of each iteration as the target estimate. Let $\bar{v}_i = (\bar{x}_{i1}, \bar{x}_{i2}, \cdots, \bar{x}_{im})$, the cluster centroid vector to which the true value v_i belongs in each iteration is $C_k^{(t)} = (c_{k1}^{(t)}, c_{k2}^{(t)}, \cdots, c_{km}^{(t)}), k = 1, \cdots, K; t = 1, \cdots, T$. We denote the Euclidean distance between vectors by $E(\cdot)$, $E(\bar{v}_i, C_k^{(t)}) = \|\bar{v}_i - C_k^{(t)}\| = \sqrt{(\bar{x}_{i1} - c_{k1}^{(t)})^2 + (\bar{x}_{i2} - c_{k2}^{(t)})^2 + \cdots + (\bar{x}_{im} - c_{km}^{(t)})^2}$ represents the Euclidean distance between $\bar{v}_i$ and the cluster centroid of the iteration t. Therefore, we can construct the objective function, the optimization process is as follows

$$
\begin{aligned}
\min\ & \textstyle\sum_{t=1}^{T} E^2(\bar{v}_i, C_k^{(t)}) \\
s.t.\ & \textstyle\sum_{j=1}^{m} (\bar{x}_{ij} - c_{kj}^{(t)}) \leq R_t^2; t = 1, \cdots, T \\
& x_{ij} \in [min(A_j), max(A_j)]; i = 1, \cdots, n; j = 1, \cdots, m.
\end{aligned} \tag{7}
$$

According to the optimization formula (7), we can calculate the feasible solution $\bar{v}_i$ satisfying the objective function and constraints. Then, we can judge whether our inference attack is successful by comparing the gap between the estimated value $\bar{v}_i$ and the original value v_i.

As Algorithm 1 below, we give an attack scheme against the LDPKmeans basic proposal, which describes in detail the specific steps of the general attack to calculate the estimated value with the smallest error from the true value. In Line 5, the user calculates the Euclidean distance of a data point from all centroids and selects the cluster index $k \in \{1, \cdots, K\}$ with the smallest distance. The user then sends the cluster information to which each data point belongs to the server. Line 7 saves cluster information about real users on the server. Line 8 is the server re-updates each cluster centroid by calculating the mean value of each cluster, and then sends the new centroid to the user. The server in Line 11 estimates

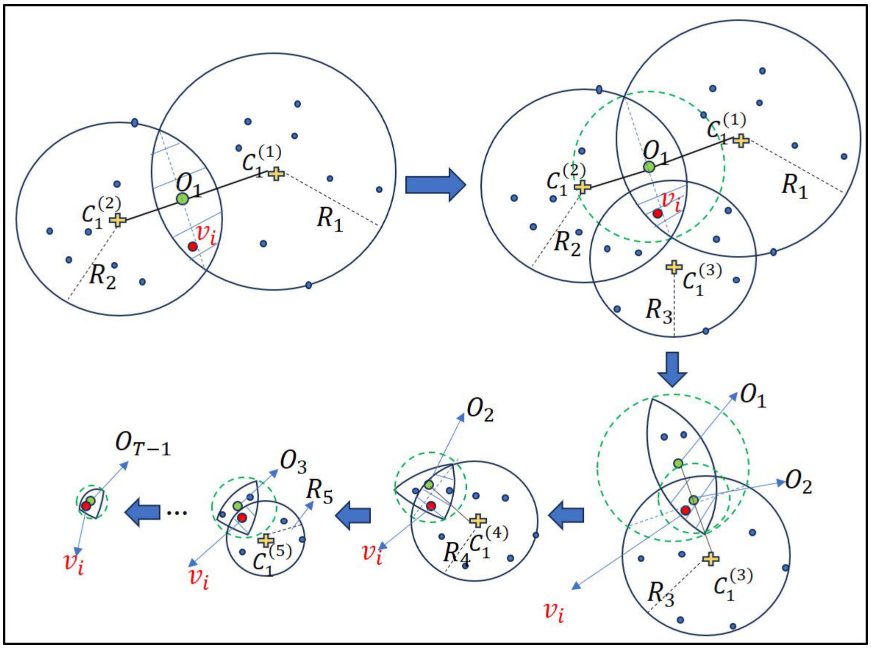


FIGURE 2. The attacker infers the user's true value v_i based on the cluster information of multiple rounds of iterations.

Algorithm 1 General Attack Scheme on LDPKmeans

```
Input: $u_i, \tilde{v}_i = (\tilde{x}_{i1}, \tilde{x}_{i2}, \cdots, \tilde{x}_{im})$
Output: Estimated value $\bar{v}_i = (\bar{x}_{i1}, \bar{x}_{i2}, \cdots, \bar{x}_{im})$
Cluster information $Z = \emptyset$;
repeat
    for $t = 1, 2, \cdots$ do
        for $i = 1, \cdots, n$ do
            User-side ← $K$ centroids $C^{(t)} = \{C_1^{(t)}, C_2^{(t)}, \cdots, C_K^{(t)}\}$ selected by the server;
            Server-side ← the user calculates $Cluster_{index} = argmin_k Dist(v_i, C_k^{(t)}), k = 1, \cdots, K$;
            $Z = Z \bigcup \{Cluster_{index}\}$, and the server groups the perturbed data $\tilde{v}_i$;
            User-side ← update each cluster center $C_k^{(t)}$;
        end for
        The server stores every cluster center $C_k^{(t)}$ and $Z$;
        Use the constraints of formula (7) to calculate the optimal estimate $\bar{v}_i$ based on the existing information $C_k^{(t)}$ and $Z$;
    end for
until The centroid of each cluster no longer changes.
return $\bar{v}_i$
```

the real value v_i by using the constraints of formula (7), so that the distance between the estimated value $\bar{v}_i$ and the centroid of each iteration is the smallest.

2) ATTACKS BY MALICIOUS SERVICE PROVIDER

In the above basic scheme, the server is honest and curious, and only faithfully records the cluster information of each round of real data in the clustering process. Then, we make a simple inference within the number of convergence rounds based on the existing information. This kind of inference often takes a long time to run and the accuracy rate is relatively average. In this section we discuss the case where the service provider is malicious, it is very intelligent. Assume that the attacker has more background knowledge of users, and can carefully design some specific points based on the information he has, so as to infer the real value v_i more accurately and efficiently.

Similarly, we give a simple example to introduce how smart malicious service providers can carefully design points to conduct inference attacks more efficiently. As shown in Fig. 3, let the total iterative round type be T, when the distributed clustering algorithm undergoes $t-1(t-1<T)$ rounds of iterations, the malicious service provider finds that data points v_1, v_5, v_8 and v_i always maintain the same cluster index according to the saved clustering information. Therefore, they believe that the range of true values can be determined more quickly based on these points. In Fig. 3, the four points v_1, v_5, v_8, and v_i are connected to each other to construct a quadrilateral shadow area smaller than the original circular area. Then the area of the shaded part is continuously reduced, and the real value v_i can be estimated more quickly and accurately in a smaller range.

At present, we construct the attack ideas of malicious service providers in detail. During the clustering process, the malicious attacker finds that the cluster information of some points is always consistent with v_i according to the cluster information of the previous $L(L<T)$ rounds of data recorded by the aggregator. According to the estimated value $\bar{v}_i$ obtained in $T=L$ rounds in the basic scheme, the malicious

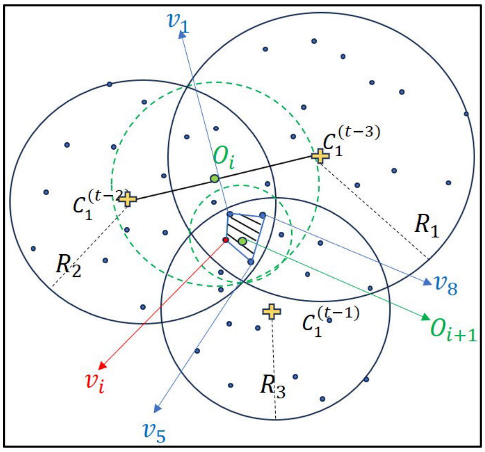


FIGURE 3. Malicious servers carefully design some specific points for reasoning attacks.

attacker carefully designs three points that are very close to $\bar{v}_i$. Let coordinate value $v_p = (x_{p1}, x_{p2}, \ldots, x_{pm})$, $v_q = (x_{q1}, x_{q2}, \cdots, x_{qm})$, $v_r = (x_{r1}, x_{r2}, \cdots, x_{rm})$, we can infer the real value according to the following optimization formula.

$$\begin{aligned}
\min\quad & [3 \cdot \textstyle\sum_{t=1}^{L} E^2(\bar{v}_i, C_k^{(t)}) - \displaystyle\sum_{a \in p,q,r} \sum_{t=1}^{L} E^2(v_a, c_k^{(t)})] \\
s.t.\quad & \textstyle\sum_{j=1}^{m} (\bar{x}_{ij} - c_{kj}^{(t)}) \leq R_t^2;\ t = 1, \cdots, L \\
& v_p, v_q, v_r \in \{\mathbb{C}_k^{(1)} \cap \mathbb{C}_k^{(2)} \cap \cdots \cap \mathbb{C}_k^{(L)}\};\ k = 1, \cdots, K \\
& x_{ij} \in [min(A_j), max(A_j)];\ i = 1, \cdots, n;\ j = 1, \cdots, m.
\end{aligned} \tag{8}$$

From formula (8), we carefully design the coordinates of three points to get the estimated value of the true value, and it is more efficient and faster. We demonstrate this conclusion experimentally in Section IV.

B. ATTACKS ON THE PRIVACY ENHANCED SCHEME OF LDPKmeans

In the privacy enhanced scheme of LDPKmeans, each iteration user u_i generate a binary string $s(|s| = K)$ and a vector $S = z, z, \cdots, v_i, \cdots, z$ related to the real data v_i. s acts as an indicator vector of information about the cluster to which a user belongs. When u_i is calculated to belong to the k-th cluster, the k-th ($k \in \{1, \cdots, K\}$) bit in s is set to 1, and the remaining $K - 1$ bits are set to 0. z in S is an all-zero string and has the same length as v_i, and the position of v_i in S represents its real cluster information. Then, each user u_i randomly perturbs each bit in the vectors S and s according to formula (4), and sends the perturbed results S' and s' to the service provider, denoted as $R_i = \{S', s'\}$. Different from the basic scheme in [5], the privacy enhanced scheme does not need to directly submit the real cluster information to the server in each iteration, but only needs to submit the set perturbed vector $\{S', s'\}$ to update the cluster information on the server. The sum of the bits of each position of s' is the estimated value $num_k$of each cluster number. The sum of bits at each position of S' is the estimated value $\widetilde{sum}^k$ of the sum of data vectors in each cluster. The service provider can then group the perturbed data and iteratively update the cluster centroids.

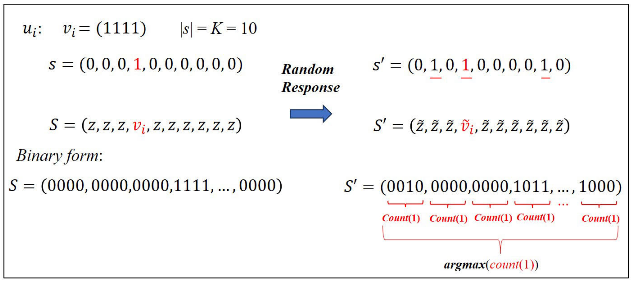

u_i: $v_i = (1111)$ $|s| = K = 10$
$s = (0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$
Random Response
$s' = (0, \underline{1}, 0, \underline{1}, 0, 0, 0, 0, \underline{1}, 0)$
$S = (z, z, z, v_i, z, z, z, z, z, z)$
$S' = (\tilde{z}, \tilde{z}, \tilde{z}, \tilde{v}_i, \tilde{z}, \tilde{z}, \tilde{z}, \tilde{z}, \tilde{z}, \tilde{z})$
Binary form:
$S = (0000, 0000, 0000, 1111, \ldots, 0000)$
$S' = (0010, 0000, 0000, 1011, \ldots, 1000)$
Count(1) Count(1) Count(1) Count(1) ... Count(1)
argmax(count(1))

FIGURE 4. An example of inferring cluster information of user u_i in LDPKmeans privacy enhancement scheme.

From the above steps, it can be seen that in each iteration, the user u_i does not directly submit the real cluster information of the data, but submits $R_i = S', s'$. S' contains the zero vector z and the perturbed data vector v_i, and the position of the perturbed data vector is the real cluster information. Since the length of each component vector is $l_j = |B_{ij}|$, we use l_j as the unit to count the number of "1" in each l_j segment in S'. Then we add the number of "1" in each unit of S' to the value (0 or 1) at the corresponding position in s' to get a $sum(1)$. Finally, we compare the index of the largest component position of $sum(1)$ to be the cluster index of v_i.

As shown in Fig.4, let the binary representation of v_i be (1111), $l_j = |v_i| = 4$. After user u_i generates vectors S and s to obtain S' and s', random perturbation is performed on each bit to obtain S' and s'. Then we take every 4 bits as a group, count the number of "1" in each group in S' and add the value at the position corresponding to s'. Finally, the group index with the largest number of "1" returned is the real cluster information to which the current user data belongs. After obtaining the cluster index of the user, we can use the attack method in Section III-A to conduct inference attacks on the original data, and finally obtain the exact value of the original data.

Algorithm 2 is a further attack step for LDPKmeans with privacy enhancement. We first deduce the user's real cluster information according to the known conditions in the LDPKmeans enhancement scheme, and then call Algorithm 1 to achieve the attack. Line 5 counts the number of "1"s in each $|v_i|$ segment in S'. Line 6 Add the j-th segment $count(1)$ of S' and the corresponding j-th element in s' to obtain the total number of "1" in each segment. These two lines are a reasoning process for the cluster information to which the real data belongs. Line 8 find the index value that makes $count(1)$ the largest in each group, that is, the corresponding cluster information of the user. After obtaining the true cluster index, Line 10 calls the general attack scheme of Algorithm 1 to infer the optimal estimate of the true value.

Algorithm 2 Enhanced Attack Scheme on LDPKmeans With Privacy Enhancement

```
Input: User $u_i, S = \{z, z, \cdots, v_i, \cdots, z\}, s(|s| = K)$ is a binary string, $z = (0, 0, \cdots, 0)$
Output: Estimated value $\bar{v}_i = (\bar{x}_{i1}, \bar{x}_{i2}, \cdots, \bar{x}_{im})$;
$Num = \emptyset$;
for $i = 1, 2, \cdots, n$ do
    $R_i = S', s' \leftarrow$ the user perturbs $S, s$;
    for $j = 1, 2, \cdots, K$ do
        $count(1) \leftarrow$ take the length of $|v_i|$ as a group in $S'$, count the number of "1"
        $Num = Num \bigcup \{count(1) + s'[j]\}$;
    end for
    $Cluster_{index} = argmax_j(Num(j))$;
end for
$\bar{v}_i \leftarrow$ Call Algorithm 1 with cluster information $Cluster_{index}$ of $u_i$
return $\bar{v}_i$
```

There are two reasons for successful attacks on privacy-enhancing schemes of improved LDPKmeans. On the one hand, in the enhancement scheme of LDPKmeans, each bit in the S' and s' vectors are required to satisfy ε-LDP, and then $R_i = S', s'$ of each user satisfies $2(\sum l_j + 1)$-LDP. The scheme satisfies $2T(\sum l_j + 1)$-LDP for T iterations. From the value of ε and the binary length l_j of the data value selected in the experiment in [5], the privacy budget of each round is very large, and the overall privacy protection ability is weak. On the other hand, according to the bit flipping probability of the random response in formula (4), each bit remains unchanged with a larger probability, and flips to the opposite value with a smaller probability. We can speculate that only the vector segment after the perturbation of the original data in S' contains the most number of "1". Therefore, as long as we find out the index of the vector segment with the largest number of "1", it is the real cluster information of the current data. In short, it can be seen that the privacy enhanced scheme in [5] still leaks user privacy.

IV. EXPERIMENT EVALUATION AND DISCUSSION

Following the description of the attack method in Section III, we demonstrate the effectiveness of our inference attack through experimental evaluation. We want to know: 1) How accurate are honest but curious service providers inferring estimates before the LDPKmeans algorithm converges; 2) How efficiently and quickly malicious service providers can infer estimates by carefully crafting a few data points.

A. EXPERIMENT SETTINGS

Data Sets: We conduct an attack experiment of K-means clustering under local differential privacy on two real data sets: The first dataset is 3D Road Network,[1] it contains 434,874 pieces of location information, 4 features*ID*, *longitude*, *latitude*, *altitude*. The second dataset is Shuttle,[2] it contains a total of 58,000 numerical data records and 9 features. Among them, the training set is 43,500 and the test set is 14,500. We select the data records of the entire test set and the top 5 features for experiments. To present the dataset more intuitively, Table 2 gives the basic information of the real dataset used in this paper.

TABLE 2. The datasets used in our experiments.

Dataset	Number of records	Number of features
3D Road Network	434,874	4
Shuttle	58,000	9

Metrics: In this experiment, we evaluate our attack scheme using two commonly used metric, relative error (RE) and root mean square error (RMSE). Both RE and RMSE can be used to measure the difference between the estimated value vector and the real value vector. The smaller the RE and RMSE, the smaller the error between the real value and the predicted value. The calculation formula of RE and RMSE are as follows:

$$RE = \frac{1}{n} \sum_i \frac{\|\bar{v}_i - v_i\|}{\|v_i\|} \tag{9}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\|\bar{v}_i - v_i\|)^2} \tag{10}$$

Comparison Method: There is an attack idea in [5], where the server still knows the precise cluster information to which each user belongs because each iteration is highly correlated with the user data. We implement this attack method and name it LDPK_Attack. We use the LDPK_Attack method as a baseline to compare with our method. The attack method in this paper is similar to the LDPK_Attack method in that both use the clustering information saved in multiple rounds of distributed k-means iterations to determine the range of true values. The novelty of our attack method lies in using optimization functions to build a suitable mathematical model and then calculating feasible solutions that satisfy the objective function and constraints.

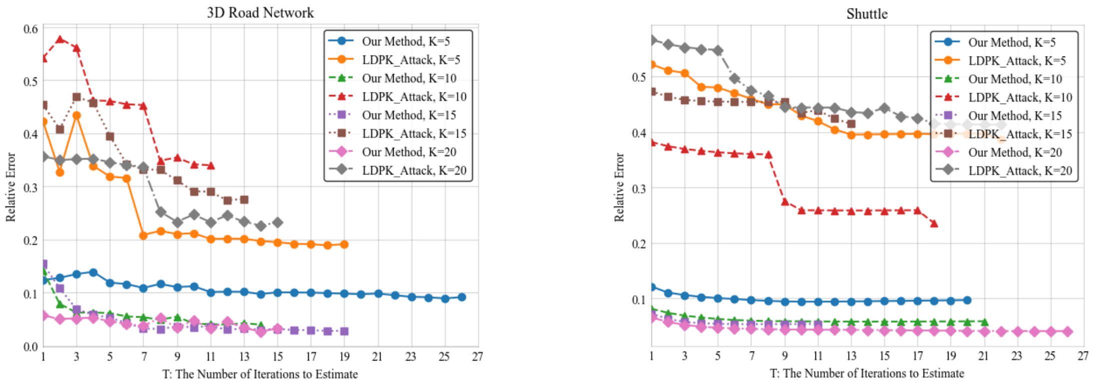
B. EXPERIMENTAL RESULTS

This experiment considers the RE and RMSE between the estimated value and the real value of our attack scheme under different iteration times T and different clustering K values (K=5, 10, 15, and 20). The abscissa in the figure represents the current number of iterations T, and the ordinate represents the error(RE or RMSE) between the estimated value and the real value under a certain number of iterations.

Fig. 5 and Fig. 6 depict the RE and RMSE results after our basic attack method and the LDPK_Attack method attack distributed k-means clustering. Among them, LDPK_Attack

[1] http://archive.ics.uci.edu/dataset/246/3d+road+network+north+jutland+denmark

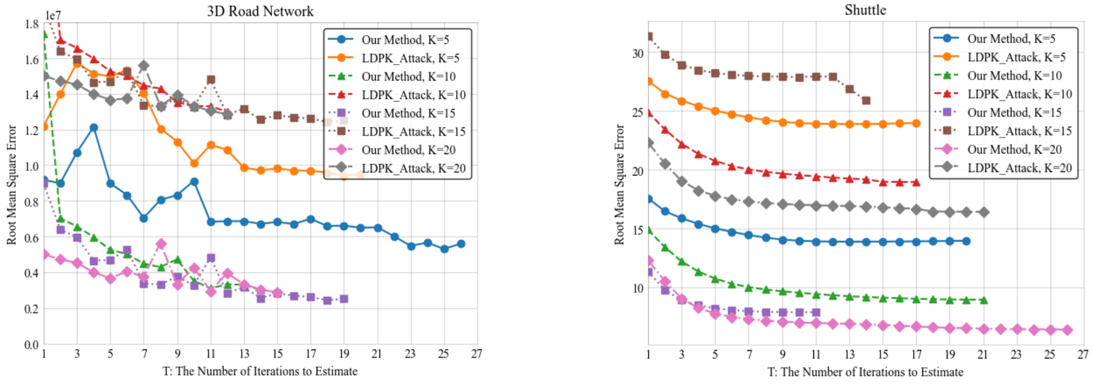
[2] http://archive.ics.uci.edu/dataset/148/statlog+shuttle



(a)

(b)

FIGURE 5. The RE of the honest service provider attack under different cluster numbers *K* on (a) 3D Road Network. (b) Shuttle.



(a)

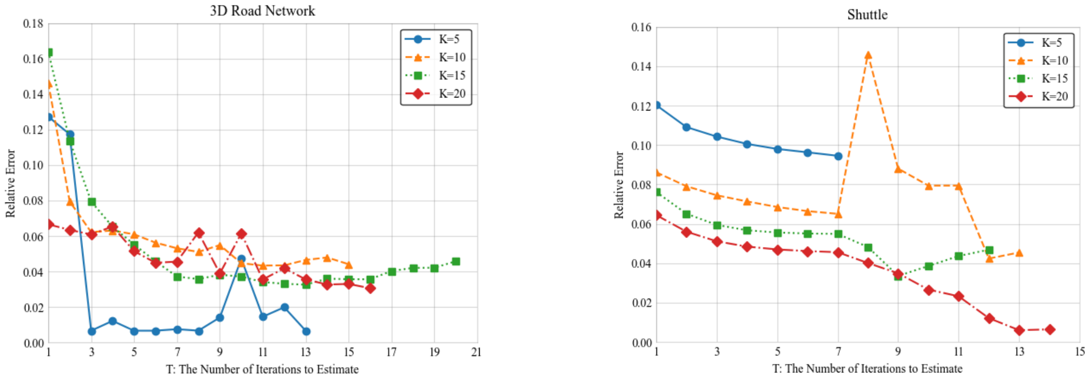
(b)

FIGURE 6. The RMSE of the honest service provider attack under different cluster numbers *K* on (a) 3D Road Network. (b) Shuttle.

is a solution we implemented based on an attack idea in [5]. Fig.7 and Fig.8 are the RE and RMSE results of the malicious service provider's attack on the LDPKmeans basic scheme. We evaluate on the real dataset 3D Road Network and Shuttle, and randomly select the estimated values of 1000 user points and average them to get the final error result. In Fig. 5 (a) and (b), the RE obtained by our method when the LDPKmeans algorithm converges is less than 0.1, while the lowest RE of the LDPK_Attack method is greater than 0.2. As the number of iterations T increases, the value of RE tends to decrease. When K=20, the overall error drops the fastest and the error is the smallest. It can be seen that our attack scheme can almost successfully infer the original data value. It can be seen that the RE of our attack method is much smaller than that of LDPK_Attack. The smaller the error, the higher the accuracy of the attack.

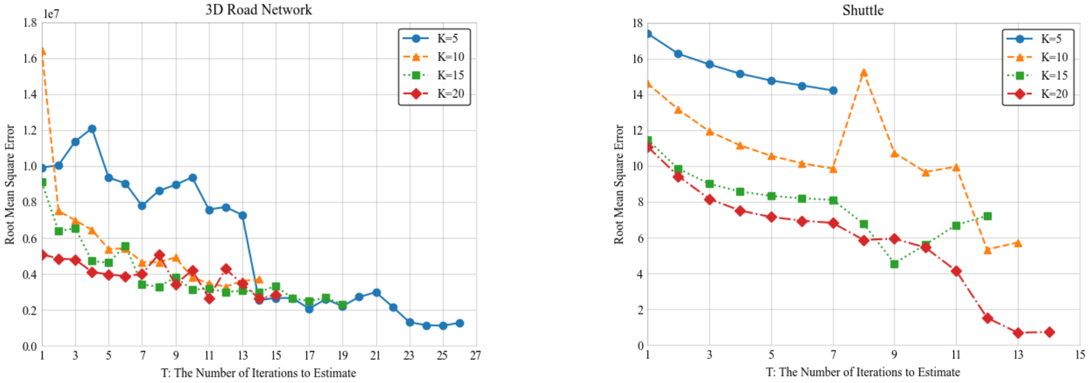
Similarly, we compare the RMSE results of our attack method and the LDPK_Attack method in Fig. 6(a) and (b). In Fig.6(a), when $K = 15$, the RMSE value achieved by our attack method when the algorithm converges is 0.27×10^7, while the RMSE of the LDPK_Attack method is approximately 1.2×10^7. In addition, on the Shuttle dataset, our attack method and LDPK-Attack achieve RMSE of approximately 6.4 and 16.3 respectively when $K = 20$ and the algorithm achieves convergence. In conclusion, RE and RMSE are relatively small, indicating that our basic scheme attack method can infer the real value.

Fig. 7 and Fig. 8 show the RE and RMSE of our enhanced attack algorithm on the malicious server on the 3D and Shuttle datasets respectively. Here, we carefully design three points on the basis of the basic attack scheme to more precisely speculate on the conditions of the true value. The enhanced attack algorithms in Fig. 7 and Fig. 8 can converge at a faster speed with different rounds of iterations, and can achieve lower errors when converging. For example, when K=5 in Fig. 7(b), the attack has been successful in T=7 rounds, and the RE is about 0.097. For K=10, there will be a short jump in RE when T=8 rounds, which means it is due to the randomness of the algorithm. In Fig. 8(a), when K=5, the total number of iterations T is the same as the basic scheme, but the RMSE value at convergence is about 0.15, which is



(a) (b)

FIGURE 7. The RE of the malicious service provider attack under different cluster numbers K on (a) 3D Road Network. (b) Shuttle.



(a) (b)

FIGURE 8. The RMSE of the malicious service provider attack under different cluster numbers K on (a) 3D Road Network. (b) Shuttle.

TABLE 3. Comparison of the average relative errors of attack schemes under K on 3D Spatial Network.

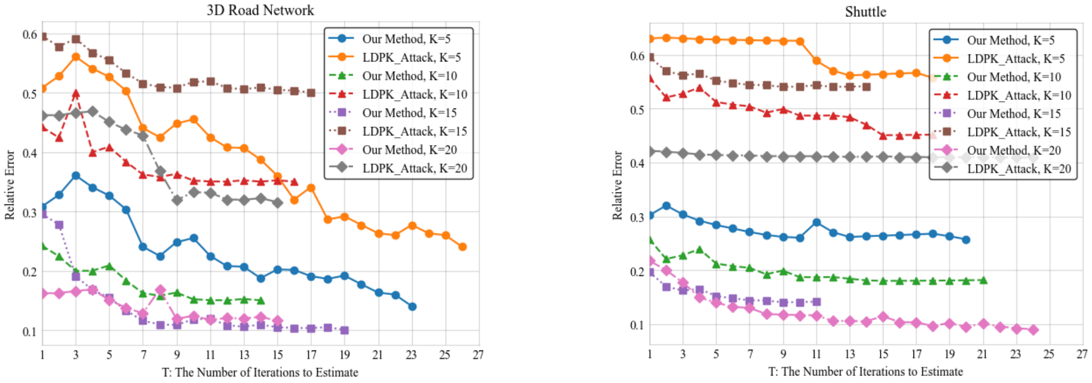
Methods	The value of K	RE(Average)
LDPK_Attack	=5	0.3761
	=10	0.3814
	=15	0.5308
	=20	0.3871
Our Method	=5	0.2381
	=10	0.1785
	=15	0.1406
	=20	0.1388

TABLE 4. Comparison of the average relative errors of attack schemes under K on Shuttle.

Methods	The value of K	RE(Average)
LDPK_Attack	=5	0.6011
	=10	0.4934
	=15	0.5519
	=20	0.4122
Our Method	=5	0.2756
	=10	0.1977
	=15	0.1548
	=20	0.1218

about 0.43 lower than the RMSE in Fig.6(a). It shows that the well-planned attack scheme is stronger and more effective than the basic scheme.

Fig. 9 shows the relative error of the attack results of our enhanced attack scheme and the LDPK_Attack scheme against the LDPKmeans privacy enhancement scheme. We use the real cluster information inference idea in Section III-B and the basic attack in Section III-A to achieve the attack on the privacy enhancement scheme. Likewise, as the number of iterations increases, RE decreases gradually. When the algorithm is close to convergence, the error is almost the smallest, indicating that our attack method is more accurate. Compared with the basic attack scheme, the RE and RMSE values in the privacy-enhanced attack scheme are larger, which is caused by the DP random perturbation in the privacy-enhanced scheme. In Fig.9(a), the RE generated by our enhanced attack scheme is about 0.14 when $K = 5$, while the RE generated by LDPK_Attack is about 0.24. Similarly, on the Shuttle data set (shown in Fig. 9(b)), when $K = 20$ and the algorithm converges, the RE of our enhanced



(a)

(b)

FIGURE 9. The RE of attacks on privacy-enhancing schemes with different numbers of *K* clusters on (a) 3D Road Network. (b) Shuttle.

attack scheme and LDPK_Attack are approximately 0.09 and 0.41 respectively.

To make the comparison clearer, Tables 3 and 4 compare the average relative errors of different attack methods on the 3D Spatial Network and Shuttle datasets, respectively. The results show that the average relative error of the true value estimated by our attack method is much smaller than that of LDPK_Attack. In addition, when the value of cluster K increases, the relative error gradually decreases. Therefore, the value of K has a certain influence on the inference of the true value. The larger the K, the more accurate the attack scheme is in deducing the true value. From a global perspective, these error values are not very large, indicating that the gap between the real value and the estimated value is not large. This further demonstrates that our attack on the privacy-enhancing clustering algorithm is successful.

In short, judging from the experimental comparison results, the attack errors (RE and RMSE) of our basic attack scheme and enhanced attack scheme are much smaller than those of the LDPK_Attack scheme. That is, our scheme can achieve higher attack accuracy than the LDPK_Attack scheme, thereby inferring more accurate original data. It further illustrates that our scheme can successfully attack distributed LDPK_means.

C. COUNTERMEASURE DISCUSSION

Our attack method fully demonstrates that distributed LDP k-means clustering can leak information through the intermediate results of its interactions, even if the real data itself is not disclosed. Therefore, a distributed clustering method with high security and efficiency is what we are looking forward to.

According to our knowledge, existing cryptographic techniques such as homomorphic encryption [11], [33] and multi-party secure computation [26] demonstrate superiority in this security performance. Homomorphic encryption is an encryption algorithm that allows non-decryption calculations on encrypted data, and the results obtained after decryption are consistent with the results of operations in plaintext. This algorithm enables k-means clustering to be implemented on untrusted third-party servers without leaking user data. Secure multi-party computation allows parties to collaboratively perform computations on their combined datasets without revealing to each other the data they possess. Therefore, we propose to use homomorphic encryption and secure multi-party computation to realize the distributed k-means clustering between the client and the service provider without revealing the real cluster information of the user.

V. RELATED WORK

Recently, the privacy-preserving problem [5], [6], [9], [11], [19], [24], [34], [35] for clustering algorithms in multi-agent systems [4], [13], [16] has been extensively studied. Reference [16] introduced encryption solutions to deal with privacy issues in multi-agent systems. Reference [6] proposed a privacy-preserving protocol for k-means clustering with arbitrary partition data distributed between two parties, which is efficient and can provide encrypted privacy protection for the data. Reference [10] created a privacy-guaranteed two-party k-means clustering protocol, an efficient way to compute multiple iterations of k-means clustering without revealing intermediate values. The method performs two-way partitioning and random uniform sampling from unknown domain sizes, and the resulting division protocol and random value protocol can be used in any protocol that requires secure computation or random sampling. Zhang et al. [11] adopted multi-key Fully Homomorphic Encryption(FHE) [33] as the main encryption primitive to securely execute multi-party k-means clustering schemes, and the entire calculation process is completely outsourced to cloud servers without leaking user data. Biswas et al. [36] proposed PPK-means, an improved k-means algorithm respecting privacy-preserving constraints. This constraint calls for devising an efficient method for estimating centroid vectors during

k-means iterations using incomplete information on binary-coded input data vectors.

Additionally, Blum et al. [37] proposed a well-known DPLloyd deployment method, which was the first to combine DP and k-means algorithm. The difference between DPLloyd and standard Lloyd is that it adds Laplacian noise in each iteration, and needs to determine the number of rounds of algorithm iterations in order to determine the noise that needs to be added in each round. However, the privacy budget of each round decreases as the number of iterations increases, resulting in severely distorted data. Su et al. [19] proposed an improved version of DPLloyd, which improved the selection of initial points for k-means clustering and designed a general framework to limit the number of iterations of the algorithm. Nevertheless, [23] pointed out that the DPLloyd has non-convergence problems, that is, it cannot guarantee to terminate at Lloyd's solution within a limited number of iterations, which will seriously affect their clustering quality and execution efficiency. To guarantee the convergence, they always keep the perturbed centroid of the previous iteration $t - 1$, calculate a convergence region for each cluster in the current iteration t, and inject differentially private noise in this region. However, these algorithms all rely on trusted third-party servers. Therefore, the differential privacy clustering algorithm for untrusted servers has attracted great attention of researchers. Xia et al. [5] proposed distributed k-means clustering under the guarantee of LDP. In this scheme, the client and the server cooperate interactively. The server is responsible for data perturbation and the calculation of the cluster information to which the user belongs, and the server performs cluster centroid selection, data grouping, and cluster centroid update. However, these k-means methods with DP guarantees, while more efficient than encryption methods, are usually not as secure or accurate.

VI. CONCLUSION

In this paper, we proposed an efficient attack method which shows LDPKmeans in [5] would seriously leak user privacy. In our proposed attack, we first proved that the real value of user agents is revealed in the interactive iteration of LDPKmeans even if the untrusted server agent only obtains the cluster information submitted by the user agents and the cluster centroid of each iteration. Then, we assumed that the untrusted server agent is malicious and can infer the real user information of the data point more efficiently and quickly through some well-designed coordinate points. Additionally, we pointed out the insecurity of improved LDPKmeans with privacy enhancement, and given an enhanced inference attack to recover the real cluster information of user agent. Theoretical analysis and experimental evaluation show that compared with existing attack methods, our attack method for distributed LDPKmeans is more efficient and can accurately infer the real data information before the algorithm converges.

In the future, we will focus on preventing information about real data from untrusted third-party servers and designing data clustering mechanism with strong security for each value.

REFERENCES

[1] J. M. Such, A. Espinosa, and A. García-Fornes, "A survey of privacy in multi-agent systems," *Knowl. Eng. Rev.*, vol. 29, no. 3, pp. 314–344, Jun. 2014.

[2] M. R. Znaidi, G. Gupta, and P. Bogdan, "Secure distributed/federated learning: Prediction-privacy trade-off for multi-agent system," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2022, pp. 97–102.

[3] Y. Yuan, W. He, W. Du, Y.-C. Tian, Q.-L. Han, and F. Qian, "Distributed gradient tracking for differentially private multi-agent optimization with a dynamic event-triggered mechanism," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 54, no. 5, pp. 3044–3055, May 2024.

[4] K. Mivule, D. Josyula, and C. Turner, "An overview of data privacy in multi-agent learning systems," in *Proc. 5th Int. Conf. Adv. Cognit. Technol. Appl.*, 2013, pp. 14–20.

[5] C. Xia, J. Hua, W. Tong, and S. Zhong, "Distributed K-means clustering guaranteeing local differential privacy," *Comput. Secur.*, vol. 90, Mar. 2020, Art. no. 101699.

[6] G. Jagannathan and R. N. Wright, "Privacy-preserving distributed k-means clustering over arbitrarily partitioned data," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2005, pp. 593–599.

[7] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Inf. Sci.*, vol. 622, pp. 178–210, Apr. 2023.

[8] D. Fiore and G. Russo, "Resilient consensus for multi-agent systems subject to differential privacy requirements," *Automatica*, vol. 106, pp. 18–26, Aug. 2019.

[9] J. Vaidya and C. Clifton, "Privacy-preserving k-means clustering over vertically partitioned data," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2003, pp. 206–215.

[10] P. Bunn and R. Ostrovsky, "Secure two-party k-means clustering," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, Oct. 2007, pp. 486–497.

[11] P. Zhang, T. Huang, X. Sun, W. Zhao, H. Liu, S. Lai, and J. K. Liu, "Privacy-preserving and outsourced multi-party K-means clustering based on multi-key fully homomorphic encryption," *IEEE Trans. Depend. Secure Comput.*, vol. 20, no. 3, pp. 1–12, Jun. 2022.

[12] O. Goldreich, *Foundations of Cryptography: Basic Applications*, vol. 2. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[13] C. X. Wang, Y. Song, and W. P. Tay, "Arbitrarily strong utility-privacy tradeoff in multi-agent systems," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 671–684, 2021, doi: 10.1109/TIFS.2020.3016835.

[14] C. Gao, D. Zhao, J. Li, and H. Lin, "Private bipartite consensus control for multi-agent systems: A hierarchical differential privacy scheme," *Inf. Fusion*, vol. 105, May 2024, Art. no. 102259.

[15] A. Chatalic, V. Schellekens, F. Houssiau, Y. A. de Montjoye, L. Jacques, and R. Gribonval, "Compressive learning with privacy guarantees," *Inf. Inference, A J. IMA*, vol. 11, no. 1, pp. 251–305, Mar. 2022.

[16] L. N. Foner, "A security architecture for multi-agent matchmaking," in *Proc. 2nd Int. Conf. Multi-Agent Syst.*, 1996, pp. 80–86.

[17] Y. Zhu, Z. Wang, and J. Wang, "Collusion-resisting secure nearest neighbor query over encrypted data in cloud, revisited," in *Proc. IEEE/ACM 24th Int. Symp. Quality Service (IWQoS)*, Jun. 2016, pp. 1–6.

[18] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[19] D. Su, J. Cao, N. Li, E. Bertino, and H. Jin, "Differentially private K-means clustering," in *Proc. 6th ACM Conf. Data Appl. Secur. Privacy*, Mar. 2016, pp. 26–37.

[20] X. Ye, Y. Zhu, M. Zhang, and H. Deng, "Differential privacy data release scheme using microaggregation with conditional feature selection," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 18302–18314, Oct. 2023, doi: 10.1109/JIOT.2023.3279440.

[21] Y. Zhu, Q. Song, and Y. Luo, "Differentially private top-k flows estimation mechanism in network traffic," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 3, pp. 2462–2472, May 2024.

[22] B. Ghazi, J. He, K. Kohlhoff, R. Kumar, P. Manurangsi, V. Navalpakkam, and N. Valliappan, "Differentially private heatmaps," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 6, pp. 7696–7704.

[23] Z. Lu and H. Shen, "Differentially private k-means clustering with convergence guarantee," *IEEE Trans. Depend. Secure Comput.*, vol. 18, no. 4, pp. 1541–1552, Jul. 2021.

[24] M. Yang, I. Tjuawinata, and K.-Y. Lam, "K-means clustering with local d_{χ}-privacy for privacy-preserving data analysis," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2524–2537, 2022, doi: 10.1109/TIFS.2022.3189532.

[25] M. Jones, H. L. Nguyen, and T. D. Nguyen, "Differentially private clustering via maximum coverage," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 13, pp. 11555–11563.

[26] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *Proc. 26th USENIX Secur. Symp. (USENIX Secur.)*, 2017, pp. 729–745.

[27] Q. Xue, Q. Ye, H. Hu, Y. Zhu, and J. Wang, "DDRM: A continual frequency estimation mechanism with local differential privacy," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6784–6797, Jul. 2023.

[28] Y. Zhu, Y. Cao, Q. Xue, Q. Wu, and Y. Zhang, "Heavy hitter identification over large-domain set-valued data with local differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 414–426, 2024, doi: 10.1109/TIFS.2023.3324726.

[29] Y. Zhang, Y. Zhu, Y. Zhou, and J. Yuan, "Frequency estimation mechanisms under $\epsilon\delta$-utility-optimized local differential privacy," *IEEE Trans. Emerg. Topics Comput.*, vol. 12, no. 1, pp. 316–327, Jan. 2024.

[30] D. Zhang, W. Ni, N. Fu, L. Hou, and R. Zhang, "Locally differentially private multi-dimensional data collection via Haar transform," *Comput. Secur.*, vol. 130, Jul. 2023, Art. no. 103291.

[31] L. Sun, J. Zhao, and X. Ye, "Distributed clustering in the anonymized space with local differential privacy," 2019, *arXiv:1906.11441*.

[32] U. Stemmer, "Locally private k-means clustering," *J. Mach. Learn. Res.*, vol. 22, no. 176, pp. 1–30, 2021.

[33] Y. Liu, Y. Luo, Y. Zhu, Y. Liu, and X. Li, "Secure multi-label data classification in cloud by additionally homomorphic encryption," *Inf. Sci.*, vol. 468, pp. 89–102, Nov. 2018.

[34] Y. Zhu, Z. Huang, L. Huang, and T. Takagi, "On the security of a privacy-preserving product calculation scheme," *IEEE Trans. Depend. Secure Comput.*, vol. 12, no. 3, pp. 373–374, May 2015.

[35] M. Shechner, "Differentially private algorithms for clustering with stability assumptions," 2021, *arXiv:2106.12959*.

[36] C. Biswas, D. Ganguly, D. Roy, and U. Bhattacharya, "Privacy preserving approximate K-means clustering," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1321–1330.

[37] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: The SuLQ framework," in *Proc. 24th ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst.*, Jun. 2005, pp. 128–138.

CONGCONG SHI received the Ph.D. degree in computer science. He is currently working as a Professor-Level Senior Engineer with the State Grid Laboratory of Power Cyber-Security Protection and Monitoring Technology, State Grid Smart Grid Research Institute Company Ltd., Nanjing, Jiangsu, China. His research interests include power system information security, data security, and industrial control security.

XIULI HUANG is currently working as a Senior Engineer with the State Grid Laboratory of Power Cyber-Security Protection and Monitoring Technology, State Grid Smart Grid Research Institute Company Ltd., Nanjing, Jiangsu, China. Her research interests include privacy computing, data security, and industrial control security.

PENGFEI YU received the master's degree from the University of Science and Technology of China. He is currently working as a Senior Engineer with the State Grid Laboratory of Power Cyber-Security Protection and Monitoring Technology, State Grid Smart Grid Research Institute Company Ltd., Nanjing, Jiangsu, China. He is also working as the Chief Engineer with the Power Grid Digitalization Institute, State Grid Smart Grid Research Institute. His research interests include data security and privacy computing.