Thwarting Longitudinal Location Exposure Attacks in Advertising Ecosystem via Edge Computing

Le Yu*‡, Shufan Zhang†‡, Lu Zhou*, Yan Meng*, Suguo Du*, Haojin Zhu*§
*Shanghai Jiao Tong University, †University of Waterloo
*{yule5100309221, zl19920928, yan_meng, sgdu, zhu-hj}@sjtu.edu.cn, †shufan.zhang@uwaterloo.ca

***Abstract*—As geo-location data has been increasingly adopted as a high-profile feature in targeted advertising, exposing user real locations to untrusted cloud services or advertisers has raised severe privacy concerns. To protect location privacy with formal guarantee, a wide-stretched line of recent studies focuses on injecting controlled geo-indistinguishability (geo-IND) noise as per each location exposure. However, in advertising, over the course of 2 years, a single user can report and contribute near 1k location data points on average, which allows a longitudinal attacker to infer some statistics from the perturbed locations.**

In this study, we demonstrate the above-mentioned privacy risk via revealing an inference attack mechanism, coined as a *longitudinal location exposure attack*. This novel attack illustrates the possibility of recovering 75%∼90% of user top-1 locations (within only 200-meter range) among 37k users. In light of this deficiency, we propose a novel edge-assisted location privacy protection system, entitled *Edge-PrivLocAd*, that is adapted to location-based advertising. The novelty of Edge-PrivLocAd stems from our n-fold Gaussian mechanism, which adds permanent noise to the statistical user location profile and thus can defend against longitudinal attackers while balancing the privacy-utility trade-off. In addition, our system incorporates a posterior-based sampling technique into the location re-mapping process, that boosts location utility without privacy loss. We develop a fully-functioning prototype and empirically evaluate the proposed system. Our experimental results show that Edge-PrivLocAd is practical and scalable in real-world scenarios.

I. Introduction

Location-based advertising (LBA) [1], [2] represents an emerging advertising technique to target mobile users, leveraging location information to decide whom to deliver ads based on where they are. Through LBA, digital marketers are benefited from a higher return on investment (ROI) since they can significantly improve the relevance of their targeted users. As Global Industry Analysts reports [3], the LBA market in the U.S. is estimated at $22.8 Billion in the year 2021, and China is forecast to reach $16.1 Billion by 2026, while the global market may reach $133 Billion by then.

Despite its advantages, this new paradigm of ads delivery raises severe privacy concerns among mobile users. In current LBA systems, users' real-time locations can be easily collected by ad service providers [2], which may be abused by them to further infer the location semantics (e.g., home and work place), mobility patterns, and even habits, interests, activities, and relationships of users. With the development of privacy protection laws such as GDPR and CCPA, users pay more attention to their location privacy. This motivates the industry to take more actions on protecting the users' privacy. For example, Apple introduces a new privacy control feature, APP Tracking Transparency, starting from iOS 14.5, which allows the users to block third-party apps and advertisers from tracking their online behaviour (e.g., their locations). However, it is reported that this feature has been bypassed by the big tech companies like Google, Meta (erstwhile Facebook), and Snap due to its challenge to the current ad ecosystem. Therefore, it is more desirable to have a user-controllable privacy-enhancing technique that is compatible with the existing LBA ecosystem.

Location privacy has been extensively investigated over the past decade. Numerous location privacy-preserving mechanisms (LPPMs) [4]–[13] have been proposed, many of which are built upon the well-known differential privacy (DP) to provide formal and rigorous guarantee on user location privacy. One mechanism, entitled geo-indistinguishability (or geo-IND) [9], has inspired a line of follow-up works applying geo-IND in different location service scenarios [10]–[12]. However, in this study, we will reveal that there still exists a huge gap between the theoretical privacy guarantee and real-world privacy issues in LBA.

More specifically, we identify a new type of attack – the longitudinal attack, aiming to infer the target user's *top locations* from his reported and obfuscated locations through long-term observation or tracking. This attack is realistic in the current ad ecosystem since any advertisers or third-party traffic verification companies can observe the location updating from the billions of ad bidding logs per day [14]. Based on the frequency of a location being reported to the ad network, the locations can be classified into two categories: *nomadic location, and top location*. The nomadic locations refer to the locations that a user rarely visits, the protection of which can be achieved by leveraging geo-IND. Different from nomadic locations, top locations are referred to as the user's most sensitive locations (home, work place, etc.), which are routinely reported and are of great value for advertising.

The protection of top locations represents a new challenge for geo-IND based location privacy protection mechanisms [10]–[12]. This is because the existing geo-IND mechanisms are based on a common assumption that every reported location is independent, which holds for nomadic locations but not for the top locations. According to the composition theorem in differential privacy [15], exposing multiple obfuscations of the same location will degrade the overall

‡ co-first authors.
§ Haojin Zhu (zhu-hj@sjtu.edu.cn) is the corresponding author.

privacy protection, which makes the protection of top locations quite different from the differential privacy perspectives. A longitudinal attacker (e.g., ad service provider or any third party observer) can thus exploit this fact to recover the real top locations once it obtains sufficient obfuscated locations. We conduct a simple but effective experiment, using 37,262 real-world LBA users' data, to illustrate that an attacker of this kind can reconstruct 80%~93% of users' top-1 locations and over 50% top-2 locations. Our attack well demonstrates the severity of this long-term exposure threat.

To defend against the longitudinal attacker, a naïve solution is to first add permanent geo-IND noise and then always report the same obfuscated location. However, the permanent noise will lead to an unacceptable degradation of utility since the location data in this solution suffer from *permanent utility loss* for every usage of user location. In LBA, an advertiser may permanently lose its potential user if this obfuscated location drops outside the targeting area and is never being updated. One mitigating factor, outputting multiple obfuscated locations instead of one, also has the drawback of the degradation of the overall privacy guarantee. Therefore, how to improve the probability of a user being targeted without compromising the privacy guarantee, especially under the longitudinal attack, still remains a big challenge, which also motivates this work.

In this paper, we propose an *edge-assisted, utility-aware* system, entitled **Edge-PrivLocAd**, which dedicates to provide privacy-preserving location data management for LBA. The novelty of the Edge-PrivLocAd system stems from the incorporation of edge devices for user location management and a new *n-fold Gaussian mechanism* to boost the LBA utility with non-degradable privacy guarantee. The trusted edge devices work as a firewall layer between mobile users and malicious service providers. We use edge devices to assess the risk of location privacy breaches, create user dynamic location statistics, and adopt the appropriate LPPM to protect user location privacy. The novel n-fold Gaussian mechanism can be regarded as a non-trivial extension to existing geo-IND mechanisms. In this mechanism, we solve the analytic challenge by introducing sufficient statistics, which provides *tighter* error bounds on noise composition when generating multiple obfuscated locations at the same time. In addition, we design a re-sampling based post-processing module to boost the utility of the output locations without privacy loss. Putting all the modules altogether, we show through formal proofs and experimental results that the Edge-PrivLocAd system is able to achieve the long-term privacy, location utility, and performance requirements.

Contributions. The main technical contributions of this paper are highlighted as follows.

- *New Attack*. We identify a novel attack, the longitudinal attack, to reveal a new location privacy threat that cannot be addressed by the existing geo-IND-based approaches. The attack is validated by using 37,262 mobile users' real-world data from the ad network.
- *New Mechanism*. We propose a novel n-fold Gaussian mechanism to defend against the longitudinal attacker and boost the utility under non-degradable privacy guarantee, which is achieved with the aid of the sufficient statistics and a re-sampling based post-processing module.
- *New System*. We design *Edge-PrivLocAd*, a full-stack edge-assisted system dedicated to providing privacy-preserving location management for LBA and reducing the computing costs of users' devices. We develop and implement a fully-functioning prototype of our system.[1] We conduct extensive analytical experiments, on a real-world advertising dataset, to well demonstrate the effectiveness and the efficiency of the proposed system.

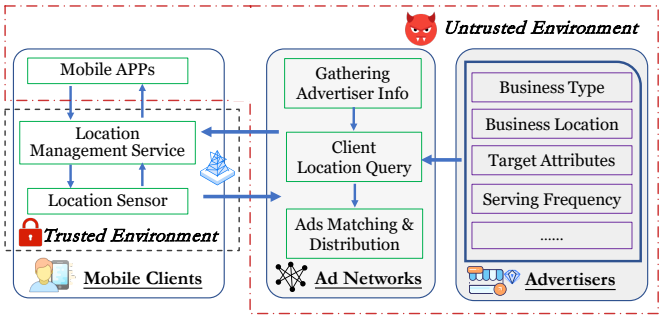


Fig. 1: The business model and data flow of LBA.

Roadmap. The remainder of this paper is organized as follows. In Section II, we introduce the preliminaries of this work. In Section III, we present the longitudinal location exposure attack faced by the geo-IND based schemes. Then, we elaborate on the design motivation of Edge-PrivLocAd in Section IV, which is followed by detailed design, privacy analysis, evaluation, and related work in Section V, VI and VII resp. Finally, we conclude this paper in Section IX.

II. Background

In this section, we first introduce the business model of LBA, as well as how advertisers can target their ads to geographic locations in real-world LBA platforms. Then we introduce geo-IND, a formal definition widely discussed to address the location privacy issues in location-based services.

A. Location-Based Advertising: Paradigm, Business Model, and Data Flow

As depicted in Fig. 1, we abstract and summarize the typical business model of current location-based advertising (LBA). The model includes three typical categories of participants: the *mobile clients* (a.k.a. *users*) who are served mobile ads on their mobile devices; the *ad networks* who match and distribute ads to mobile clients according to advertiser requirements and user attributes; and the *advertisers* (a.k.a. *vendors* or *businesses*) who want to promote their businesses to their potential customers.

Fig. 1 also illustrates more details about the work flow of LBA. To set up a location-based advertising campaign, the advertiser needs to pinpoint her business location and specify

[1]The source code is available at GitHub.

TABLE I: Targeting Range on Top Players' LBA Platforms

Companies	Minimal Radius	Maximal Radius
Google	5 km	65 km
Microsoft	1 mile / 1 km	800 miles / 800 km
Facebook	1 mile	50 miles
Tencent	500 m	25 km

a range to define the targeting area. When a user triggers an ad request, her present location will be sent to the **ad network**. The ad network then informs the advertiser to bid on the ad request whose targeting location matches the user. Once the matching is completed within a given time limit (e.g., 100 ms in [16]), this kind of personalized ads will be sent to the user device and displayed on the banner or in a pop-up manner.

Categories of location targeting. We investigate some top players' LBA platforms and most of them provide the following three categories of geo-targeting methods:

- **Countries Targeting.** Advertisers target their advertisements to a country or countries.
- **Areas Targeting.** The advertiser targets their advertisements to areas in a country, usually specified by cities or administrative areas.
- **Radius Targeting.** The advertiser submits a business location and a radius to show their advertisements to the customers within the radius from the business location. For instance, Google Ads platform [17] allows advertisers to target up to 1000 location concurrently, each of which is specified by a different center with different radius.

In this work, we focus on the radius targeting, which is widely adopted by most LBA platforms and is most privacy sensitive among the three categories of geo-targeting. In radius targeting, the user has to provide her precise location to the LBA service provider, and the service provider matches with the advertiser if their distance is within a predefined radius. The allowed targeting range differs according to different LBA platforms. We investigate LBA platforms of four famous companies and summarize the radius range as shown in Table I. It shows that the targeting radius can be as quite small, which poses a risk to the targeted users' location privacy.

B. Privacy Notion: Geo-indistinguishability

To address the location privacy issue and provide a rigorous location privacy guarantee, a formal privacy notion for location-based services named *geo-indistinguishability (geo-IND)* [9] is proposed. Geo-IND is an extension to the well-known concept of *differential privacy*. The idea of geo-IND is to enforce similar distribution between the obfuscation of any two real locations whose distance is within r, so that they can not be distinguished by any informed adversaries observing the obfuscated location. Specifically, a user can customize her privacy requirements by a tuple (l, r), where r is the radius she is mostly concerned with and l is the privacy level. This requirement can be achieved through ε-geo-IND for $\varepsilon = l/r$.

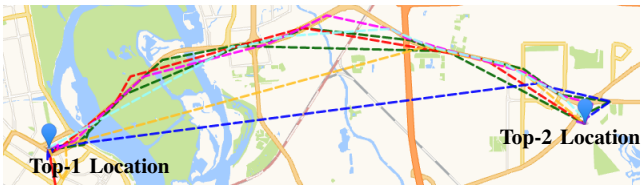


Fig. 2: A user's 7-day mobility pattern.

Definition 1 (ε-geo-IND [9])**.** *For all locations $\boldsymbol{p}_0$, $\boldsymbol{p}_1$, let $d(\boldsymbol{p}_0, \boldsymbol{p}_1)$ be their distance, a privacy mechanism $\mathcal{M}$ satisfies ε-geo-indistinguishability if for all output location $\boldsymbol{q}$:*

$$Pr[\mathcal{M}(\boldsymbol{p}_0) = \boldsymbol{q}] \leq e^{\varepsilon d(\boldsymbol{p}_0, \boldsymbol{p}_1)} Pr[\mathcal{M}(\boldsymbol{p}_1) = \boldsymbol{q}] \quad (1)$$

Geo-IND is widely discussed in location-based services and spatial crowdsourcing. In location-based services, LP-Guarding [10], LP-Doctor [11] and Top-K Geo-Query system [12] are proposed to provide obfuscated locations for point of interest (POI) queries. In spatial crowdsourcing, previous works [18], [19] propose a geo-IND based mechanism to obfuscate users' real-time location. However, none of these works have been aware that the users' mobility patterns are repeated day by day, and under one-time geo-IND mechanism, privacy levels will decrease along with the increase of the number of observations according to the composition theorem. In this study, we reveal that the user's location privacy is vulnerable under the longitudinal observation (i.e., *longitudinal location exposure attack* in the following section) from the attacker.

III. Longitudinal Location Exposure Attack

The emergence of LBA raises severe threats to users' location privacy. Although traditional geo-IND based solutions obfuscate the location information provided by users, in this section, by proposing the *longitudinal location exposure attack*, we reveal the location privacy caveat of LBA in the wild.

A. Attack Model and Goal

For an LBA system which deploys geo-IND based location privacy-preserving mechanisms, we consider a longitudinal attacker who can track the victim's real-time locations from ad networks. We argue this is possible in current ecosystem, since advertisers or ad-networks usually require user's IDs (e.g., android ID in Android OS and IDFA in IOS). The longitudinal attacker's goal is to build the victim's location profile to compute top locations and reveal mobility patterns.

Fig. 2 depicts a concrete example in the longitudinal location exposure attack, where a victim's 7-day traces with 2,414 raw spatiotemporal data are presented. In this study, we refer the raw spatiotemporal data as the term *check-in*. It is observed from Fig. 2 that the user's top locations as well as the location semantics (e.g., home and office) and the mobility patterns are not difficult to infer from the illustration.

B. Procedures of Longitudinal Location Exposure Attack

The proposed longitudinal location exposure attack consists of the following two steps.

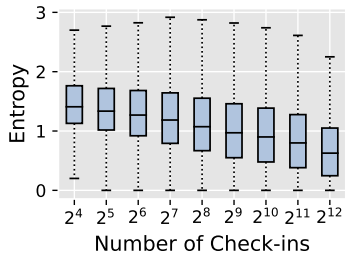
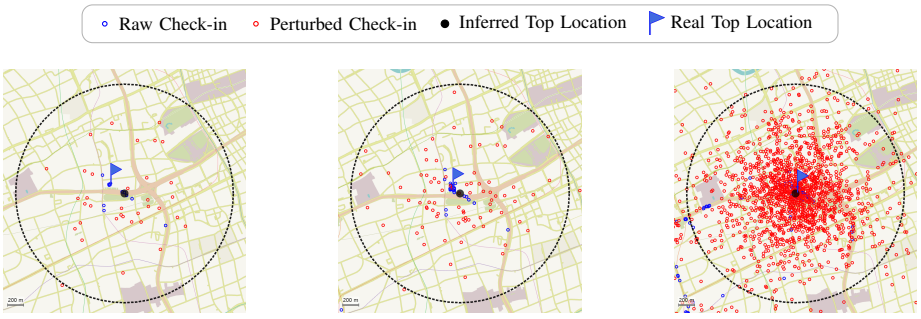


Fig. 3: Location entropy will decrease with the number of check-ins.



(a) One-week Data (b) One-month Data (c) Full Year Data

Fig. 4: An illustration of de-obfuscation attack.

1) Location Profiling Attack: With the spatiotemporal data, the attacker can build the victim's location profile and compute the top locations accurately. First, for a given user, we define her location profile $\mathcal{P}$ as a set of location and frequency tuples:

$$\mathcal{P} = \{(l_1, f_1), \ldots, (l_M, f_M)\} \tag{2}$$

where M is the number of locations and f_i is the frequency of the i-th location l_i. In this attack, we want to reconstruct user location profile from the check-ins, wherein the key challenge is that the user check-ins are *ad hoc* and cannot be directly used as real locations. We observe that users move around the locations and the check-ins are distributed in a certain range around every location. Thus, we use a clustering-based method to aggregate check-ins that are inferred to belong to the same location and estimate the frequency to build the profile. In particular, we propose a connectivity-based clustering algorithm, where we say two check-ins are *connected* if their *Euclidean* distance is within a predefined threshold (50 m in our experiment). We calculate the centroid as the location coordinate and the size of each cluster as the frequency, and build up the profile by repeating this process.

One interesting evidence, to measure whether a user's mobility pattern is stable, is the *location entropy* metric. Formally, the location entropy is defined as

$$Entropy = \sum_{i=1}^{M} \frac{f_i}{sum} \log \frac{sum}{f_i} \tag{3}$$

where *sum* represents the total number of check-ins and f_i is the frequency of the i-th location. Leveraging user's location profile, we can compute the location entropy of each user. Fig. 3 illustrates the location entropy of 37,262 mobile users from the dataset described in Section VII. It is observed that the users' location entropy declines with the increase of the number of check-ins. Furthermore, there are 88.8% of users whose location entropy is less than 2, which means in our dataset, *most users' daily activities are refrained to their top locations*.

2) De-obfuscation Attack under One-time Geo-IND: Location obfuscation is widely adopted with the geo-IND notion [9] to provide a rigorous guarantee to user location privacy. However, existing geo-IND applications [10]–[12], [18], [19] only consider one-time obfuscation. To stress the difference, we name the original geo-IND as *one-time geo-IND* and show that one-time geo-IND mechanisms are prone to attacks under the *longitudinal attacker assumption* in our scenario.

Algorithm 1 Top-n Location De-obfuscation Attack

Input: User's obfuscated check-ins $x = \{x_1, x_2, \cdots, x_n\}$; Distance threshold θ; Cluster radius r_α;
Output: Top-n locations;

procedure INFERENCE(x)
  $connectivity(x_i, x_j) \triangleq dist(x_i, x_j) < \theta$
  for $i \in [1, n]$ **do**
    $clusters \leftarrow$ Cluster x based on connectivity
    $C \leftarrow$ The largest cluster in $clusters$
    TRIMMING(C)
    $centroid \leftarrow$ The centroid of C
    $x \leftarrow x - C$ ▷ Remove the clustered points
    yield Top-i = $centroid$
procedure TRIMMING(C: the input cluster)
  repeat
    $centroid \leftarrow$ The centroid of C
    for each $point \in C$ **do**
      if $dist(point, centroid) > r_\alpha$ **then**
        Remove $point$ from C
    for each $point \in x$ **do**
      if $dist(point, centroid) < r_\alpha$ **then**
        Add $point$ to C
  until No more points to update

The attack is set up as follows. In the one-time geo-IND obfuscation mechanism, we add independent noises to every check-ins to mimic the scenario where obfuscated locations are used in LBA services. We adopt the planar Laplace mechanism and set the privacy parameters consistent with the original geo-IND paper [9]. We propose our de-obfuscation algorithm as described in Algorithm 1 to infer the top locations. Note that obfuscated check-ins from different locations may not be separable. To tackle this challenge, our algorithm combines two stages to use clustering and trimming methods respectively. In the first stage, we use the described connectivity-based clustering method (Alg. 1: 2) to find the largest clusters. If two

check-ins are connected, we merge them into the same cluster (Alg. 1: 4-5). In the second stage, we use a trimming method to optimize the cluster (Alg. 1: 6). Specifically, we first define the radius r_α of the cluster. For each step, we discard those check-ins whose distances from the centroid are larger than r_α and update the centroid (Alg. 1: 12-15). Then we join new points into the cluster. We update the cluster until no locations should be discarded or added to the cluster (Alg. 1: 16-18). To define r_α, we consider a confidence level α where

$$\Pr[dist(\boldsymbol{p}, \boldsymbol{q}) > r_\alpha] \leq \alpha \tag{4}$$

which means the obfuscated check-in whose distance is larger than r_α is almost impossible and should be discarded. We use $r_{0.05}$ as our cluster radius. When we compute a previous top location and want to compute the next one, we will remove the check-ins in the previous cluster and repeat our algorithm again (Alg. 1: 7-9).

C. A Case Study of the Longitudinal Location Exposure Attack

We illustrate an example of our attack in Fig. 4, with the victim containing 1,969 check-ins in a year, including 1,628 top-1 check-ins. We use the geo-IND mechanism proposed in [9] to obfuscate every check-ins and then use our de-obfuscation attack to recover the top-1 location. To further evaluate the performance of our attack in different length of time window, we conduct our attack in terms of one week (Fig. 4a), one month (Fig. 4b) and full year (Fig. 4c). The results show that with longer time window, the attacker can recover the user's real location more accurately (the inference distance is 200 m in one-week attack compared to less than 50 m in full-year attack).

IV. Motivation of Edge-PrivLocAd

After the demonstration of the longitudinal attack that can accurately recover user top locations from long-term observations, in this section, we propose a novel edge-assisted location privacy-preserving system named Edge-PrivLoaAd. First, we formulate the problem of protecting user location privacy in LBA settings. Then, we reason the privacy and utility definitions considered in Edge-PrivLocAd. Finally, we discuss the design principals and goals of our Edge-PrivLocAd system.

A. Location Privacy Preserving in LBA under the Existence of Longitudinal Attacker

Considering the edge-computing based LBA system that includes **3 entities**, i.e., the mobile users, the edge devices, the LBA service providers (w.l.o.g., this includes ad networks and advertisers), we assume the mobile users and the edge devices are *trusted* participants in the system whereas the service provider is *honest-but-curious*. That is to say, the service provider honestly follows the protocol but is curious about obtaining the users' real locations for reasons such as improving the ad matching model. We additionally assume that the edge devices are maintained in the public interest and there is no collusion between the edge devices and the *honest-but-curious* entities.

We model the longitudinal privacy threat as a parameter estimation problem. As an *honest-but-curious* attacker, the LBA service provider can collect the user's reported locations over the ad network, whose goal is to infer one of the top locations (e.g., home, work place). Let the observed locations be $\mathbb{Q} = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n\}$. For the targeted top location, we assume the attacker has some prior information about the victim's possible locations $\mathbb{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k\}$, which are within certain range of r from the victim's real location. Now the attacker tries to further infer which one is the victim's real location, and the inference is essentially a parameter estimation problem given the observed locations (reported from the users/edge devices) which can be modeled as:

$$\hat{\boldsymbol{p}} = \arg\max_{\boldsymbol{p} \in \mathbb{P}} Pr[\boldsymbol{p} | \boldsymbol{q}_1, \ldots, \boldsymbol{q}_n] \tag{5}$$

To defend against the attack, a *location privacy-preserving mechanism* (LPPM) must be enforced on the trusted environment (i.e., user local side and the edge side) in the system. Every time an LBA service is triggered, an obfuscated location is generated from the trusted environment and sent to the LBA service provider.

B. Privacy and Utility Definitions in Edge-PrivLocAd

We extend the privacy notion of *geo-IND* to a new variant that the LPPM can generate a set of multiple obfuscated locations *at the same time*. For the ease of analytics, we consider the *bounded version* of *geo-IND*, which introduces a small value δ in the inequality to allow a negligible exception. This relaxation allows us to introduce other distributions than Laplacian from the location-scale family and new analytic tools to prove the satisfaction of the privacy definition.

Definition 2 (r**-Neighbouring**)**.** *For all pair of locations $\boldsymbol{p}_0$, $\boldsymbol{p}_1$, we say $\boldsymbol{p}_0$, $\boldsymbol{p}_1$ are r-neighbouring if the Euclidean distance between $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ is less than r, that is $dist(\boldsymbol{p}_0, \boldsymbol{p}_1) < r$.*

Definition 3 ($(r, \varepsilon, \delta, n)$**-geo-IND**)**.** *An* LPPM *satisfies $(r, \varepsilon, \delta, n)$-geo-indistinguishability, if $\forall \boldsymbol{p}_0, \boldsymbol{p}_1$ that are r-neighbouring and for all set of output locations $\mathbb{Q} = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n\}$, the following inequality holds,*

$$Pr[\text{LPPM}(\boldsymbol{p}_0) = \mathbb{Q}] \leq e^{\varepsilon} Pr[\text{LPPM}(\boldsymbol{p}_1) = \mathbb{Q}] + \delta \tag{6}$$

We define two utility metrics in terms of a given targeting radius R. From the user's perspective, the relevant advertisers are those whose targeted locations are within the circle of radius R from the user. We define this circle as the *area of interest* (AOI). However, due to the location obfuscation mechanism, the circle will be shifted to the obfuscated locations where the ads are actually requested and which we define as *area of request* (AOR). Now we define two utilities with respect to AOI and AOR. We first define *utilization rate* to quantify how many locations in AOI will not drop outside the shifted circle AOR.

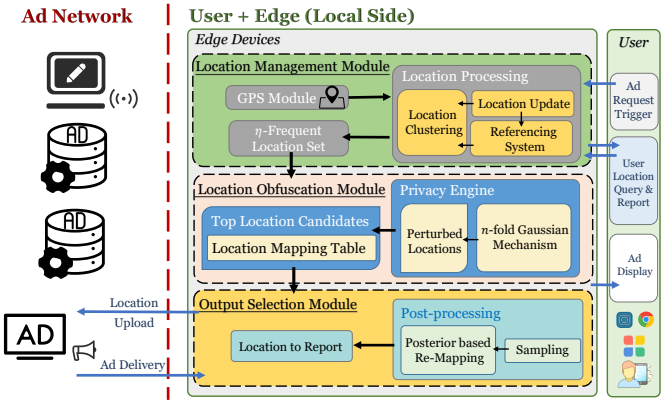


Fig. 5: The architecture of Edge-PrivLocAd.

Definition 4 (**Utilization Rate**). *The utilization rate is the proportion of AOI that are overlapped with AOR.*

$$UR = \frac{AOI \cap AOR}{AOI} \tag{7}$$

We also define efficacy to measure the probability an ad sent to the user is relevant to the user's true location.

Definition 5 (**Advertising Efficacy**). *The advertising efficacy is the probability of an ad being in the range of AOI when it is requested from an AOR.*

$$AE = Pr[ad \in AOI \mid ||ad \in AOR||] \tag{8}$$

C. Design Goals of Edge-PrivLocAd

In this study, we present Edge-PrivLocAd, a novel edge-assisted system for mobile users to preserve location privacy in LBA services. Edge-PrivLocAd conforms with the following three design goals.

Provable protection against longitudinal attackers. The proposed system should be able to defend against the aforementioned longitudinal attacker. Formally, it is required to satisfy the $(r, \varepsilon, \delta, n)$-geo-IND definition.

Optimized location utility for advertising. The proposed system should include specific techniques to improve the utility requirements in LBA, i.e., the utilization rate and advertising efficacy.

Scalable and practical edge-assisted system. The proposed system introduces edge devices as featured participants to enhance the performance and scalability of the system and reduce the computational burden on the mobile client-side.

V. System Flow of Edge-PrivLocAd

A. System Overview

In order to prevent the longitudinal attackers from de-obfuscating the top locations, we propose a system which relies on edge devices to act as data brokers which provide obfuscated top locations for long-term usage. The role that the edge devices play in our system is three-fold. First, edge devices provide services to nearby mobile users whose locations are closely distributed, making ad requesting process efficient. Second, for users with multiple mobile devices, the edge devices can provide an integrated obfuscation to prevent the degradation of privacy level further. Third, since the obfuscation mechanism will retrieve irrelevant ads, the edge devices can filter out the irrelevant ads and return the user devices with clean data, which can reduce the bandwidth overhead.

The Edge-PrivLocAd includes three modules: location management module, location obfuscation module, and output selection module. The workflow between these modules is illustrated in Fig. 5. Specifically, when a user makes an LBA request, she first sends her present location to the edge devices. The first module collects the location data to build and manage her location profile. When it reaches a time window, it will compute her top locations and send them to the second module. In the second module, it generates a set of obfuscated locations for every top location, and renders these locations as candidate outputs for LBA requests. At the same time, the output selection module draws a location from the candidate locations constructed previously to replace the true location for the LBA request. When the LBA system returns the ads to the edge device, it will select those whose locations are within the AOI to send to the target user.

Algorithm 2 Compute η-Frequent Location Set

Input: An ordered sequence $\mathcal{P} = \langle (l_1, f_1), \ldots, (l_M, f_M) \rangle$ where $f_i \geq f_j$ if $i > j$; Threshold η
Output: A set of top locations $\mathbb{T}$

```
total_freq = 0
for (l, f) ∈ P do
    total_freq += f
    Add l to T
    if total_freq ≥ η then
        return T
```

B. Location Management Module

The objective of this module is to manage the user's location profile to compute the top locations for obfuscation. It first collects location check-ins passively when user send an LBA request. Then, it periodically computes the top locations according to a configurable time window to construct the set of top locations in this period. This set is constructed periodically since users will possibly (although not frequently) change their top locations in real life. We model the top location set as the *η-Frequent Location Set* in Definition 6, which represents the most frequent locations the user appears in.

Definition 6 (η-Frequent Location Set). *Let the sequence $\mathcal{P} = \langle (l_1, f_1), \ldots, (l_M, f_M) \rangle$ be user's location profile where $f_i \geq f_j, \forall i > j$. η-frequent location set is the minimal set such that the sum of top k frequencies is no less than η.*

$$\mathrm{L}_\eta = min\{l_1, \ldots, l_k | \sum_{i=1}^{k} f_i \geq \eta\} \tag{9}$$

Since users may access different edge devices at different locations, the edge devices can only record a local part of

Algorithm 3 Obfuscation Mechanism

Input: Standard deviation σ; Real location (x, y)
Output: A set of obfuscated locations $\{(x'_1, y'_1), \ldots, (x'_n, y'_n)\}$
1: **for** $i \in 1 \ldots n$ **do**
2: Draw θ uniformly in $[0, 2\pi)$
3: Draw s uniformly in $[0, 1)$
4: Compute $r = F_R^{-1}(s)$
5: $x'_i = x + r\cos\theta, y'_i = y + r\sin\theta$
6: **return** $\{(x'_1, y'_1), \ldots, (x'_n, y'_n)\}$

the whole location profile, and finally merge the η-Frequent Location Set. For privacy considerations, this step can be accomplished through a secure multi-party computation protocol, which is however orthogonal to this work.

C. Location Obfuscation Module

The system maintains an obfuscation table $\mathcal{T}$, which maps every top location to its obfuscated version. For each top location, it will generate a set of obfuscated locations rather than a single one. The benefit comes from the fact that generating multiple obfuscated locations can improve the utilization rate. Once a new η-Frequent Location Set is constructed after a time window, the location obfuscation module starts to obfuscate the top locations from the set. For every top location, the module first checks whether it has already been obfuscated in table $\mathcal{T}$. If not, it then uses the following LPPM to generate a set of obfuscated locations and permanently records them in the table $\mathcal{T}$.

We propose an n-fold Gaussian mechanism as our LPPM, which randomly generates n obfuscated locations simultaneously given a real location. The randomness provides necessary obfuscation to the real location to achieve $(r, \varepsilon, \delta, n)$-geo-IND. As the name indicates, our obfuscation mechanism is based on Gaussian distribution. The Gaussian mechanism was first introduced as a differential privacy mechanism in [15], which adds noise drawn from a Gaussian distribution with its variance calibrated according to the sensitivity and privacy parameters. In our n-fold Gaussian mechanism, we compute the obfuscated locations by adding the real location $\boldsymbol{p}$ with n independent noises drawn from the Gaussian distribution. Specifically, the mechanism is defined as follows:

Definition 7 (n-fold Gaussian Mechanism)**.** *Given a real location $\boldsymbol{p}$, the n-fold Gaussian* LPPM *is a random sampler:*

$$\text{LPPM}(\boldsymbol{p}) = (\boldsymbol{p} + X_1, \ldots, \boldsymbol{p} + X_n) \tag{10}$$

where X_i are i.i.d. random variables from $\mathcal{N}(0, \sigma^2)$

To achieve $(r, \varepsilon, \delta, n)$-geo-IND, we set the σ of the n-fold Gaussian Mechanism as the following equation, and leave the proof of the equation in Theorem 2.

$$\sigma = \frac{\sqrt{n}r}{\varepsilon}\sqrt{\ln\frac{1}{\delta^2} + \varepsilon} \tag{11}$$

The next problem is how to compute the obfuscated locations. Our algorithm first independently samples n Gaussian noises from $\mathcal{N}(0, \sigma^2)$, and adds the noises to the real location $\boldsymbol{p}$ to get n obfuscated locations. Remember our noises are identically sampled from the same probability distribution, thus we only discuss how to sample a single random noise. To sample the noise, we consider the probability density function in polar coordinates:

$$f(r, \theta) = \frac{1}{2\pi\sigma^2} r e^{-\frac{r^2}{2\sigma^2}} \tag{12}$$

The polar coordinate system brings us convenience in that we can independently sample a radius r and an angle θ, since their marginal distributions can be easily computed as:

$$f_R(r) = \int_0^{2\pi} f(r, \theta) d\theta = \frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}} \tag{13}$$

$$f_\Theta(\theta) = \int_0^{+\infty} f(r, \theta) dr = \frac{1}{2\pi} \tag{14}$$

From the marginal density function, we see the θ is uniformly distributed from the interval $[0, 2\pi)$, which is easy to sample. Meanwhile, to sample the radius r, we need to compute the cumulative distribution function of r:

$$F_R(r) = \int_0^r f_R(\rho) d\rho = 1 - e^{-\frac{r^2}{2\sigma^2}} \tag{15}$$

We can uniformly sample a random number s from the interval $[0, 1)$, and compute the inverse function $r = F_R^{-1}(s)$ to obtain the random variable r. Finally, the coordinates of the obfuscated location is computed as

$$x' = x + r\cos\theta, \;\; y' = y + r\sin\theta \tag{16}$$

where, x, y is the coordinates of real location. The obfuscation algorithm is summarized in Algorithm 3.

Algorithm 4 Output Selection Algorithm

Input: A set of top locations $\mathbb{T}$; Candidate locations $\mathcal{T}(t) = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n\}$
Output: Obfuscated location $\boldsymbol{q}_i$
1: Select a top location t
2: Compute the centroid $(\bar{x}, \bar{y}) \leftarrow \mathcal{T}(t)$
3: Compute the density $f(x_1, y_1), \ldots, f(x_n, y_n) \leftarrow \mathcal{T}(t)$
4: Compute the probability $Pr[\mathcal{A} = \boldsymbol{q}_i] = \frac{f(x_i, y_i)}{\sum_k f(x_k, y_k)}$
5: Sample $\boldsymbol{q}_i$ with probability $Pr[\mathcal{A} = \boldsymbol{q}_i]$

D. Output Selection Module

The n-fold Gaussian mechanism can generate multiple obfuscated locations to improve the utilization rate, but the cost is the possibility of retrieving irrelevant ads. As equation (11) indicates, the magnitude of noise will increase with the number of obfuscated locations, which means the obfuscated locations will bring more irrelevant ads to the user. To reduce the overhead, we design an output selection module to select a location from the set of obfuscated locations based on Bayesian posterior probability to achieve better efficacy.

The output selection module implements the algorithm 4 to draw a location from the candidate outputs according to the posterior probability of the real location given the obfuscated

locations $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$. The posterior density function of the real location is

$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-\bar{x})^2+(y-\bar{y})^2}{2\sigma^2}} \tag{17}$$

where $\bar{x} = \frac{1}{n}\sum_k x_k, \bar{y} = \frac{1}{n}\sum_k y_k$. It quantifies the posterior probability of the real location at (x, y) given the obfuscated locations. Specifically, the density $f(x_i, y_i)$ quantifies the possibility of the real location is just placed at (x_i, y_i). Thus, we can draw every candidate location $\boldsymbol{q}_i$ with probability proportional to the following equation:

$$Pr(\mathcal{A} = \boldsymbol{q}_i) = \frac{f(x_i, y_i)}{\sum_k f(x_k, y_k)} \tag{18}$$

Once the algorithm $\mathcal{A}$ selects an obfuscated location, the system will use this location to request an ad.

VI. Privacy Analysis

In this section, we introduce a new privacy analysis tool, the sufficient statistic to prove the obfuscation mechanisms in V-C satisfy $(r, \varepsilon, \delta, n)$-geo-IND defined in Definition 3, which can inject less noise to achieve the same level of privacy compared to the composition theorem.

Sufficient statistics. We use a new privacy analysis tool, the sufficient statistic to prove the satisfaction of the $(r, \varepsilon, \delta, n)$-geo-IND when simultaneously generating multiple obfuscated locations. The sufficient statistic is a class of statistics that summarizes the samples without loss of information about the parameters to estimate. We say a statistic $T = T(X_1, \ldots, X_n)$ is sufficient if the distribution of r.v. $X_1, \ldots, X_n$ conditioned on $T = t$ does not depend on θ for all t.

In our LPPM scenario, we use $\mathrm{LPPM}(\boldsymbol{p})$ to denote the random variables from the distribution taking real location $\boldsymbol{p}$ as its parameter. Now the statistic $T = T(\mathrm{LPPM}(\boldsymbol{p}))$ is sufficient for $\boldsymbol{p}$, if for all set of generated locations $\mathbb{Q}$, the following conditional probability is independent of $\boldsymbol{p}$.

$$Pr[\mathrm{LPPM}(\boldsymbol{p}) = \mathbb{Q} | T = t] = \frac{Pr[\mathrm{LPPM}(\boldsymbol{p}) = \mathbb{Q}]}{Pr[T(\mathrm{LPPM}(\boldsymbol{p})) = t]} \tag{19}$$

With this property, we can build the connection between the sufficient statistics and the output set of the LPPM. The following theorem shows the necessary and sufficient condition for achieving $(r, \varepsilon, \delta, n)$-geo-IND.

Theorem 1. *Let the* LPPM *be a randomized mapping whose outputs are the random variables which we denote as* $\mathrm{LPPM}(\boldsymbol{p})$. $T = T(\mathrm{LPPM}(\boldsymbol{p}))$ *is a sufficient static. Then the following two statements are equivalent:*

(a) Releasing the outputs $\mathrm{LPPM}(\boldsymbol{p})$ *is* $(r, \varepsilon, \delta, n)$*-geo-IND.*

(b) Releasing $T = T(\mathrm{LPPM}(\boldsymbol{p}))$ *is* $(r, \varepsilon, \delta, n)$*-geo-IND.* [2]

Proof. (a) $\Rightarrow$ (b) immediately follows the post-processing theorem [15]. We only need to prove (b) $\Rightarrow$ (a) .

For $\forall \mathbb{Q} \in Range(\mathrm{LPPM})$, let $t = T(\mathbb{Q})$. If (b) is satisfied, we have

$$Pr[T(\mathrm{LPPM}(\boldsymbol{p}_0)) = t] \leq e^{\varepsilon} Pr[T(\mathrm{LPPM}(\boldsymbol{p}_1)) = t] + \delta \tag{20}$$

[2]We note that this theorem can be generalized to (ε, δ)-differential privacy, which may be of independent interest.

Since T is a sufficient statistic, the conditional probability in (19) is independent of $\boldsymbol{p}$ which we can denote as the function $h(\mathbb{Q}; t)$. Thus we conclude with the following inequality:

$$\begin{aligned} Pr[\mathrm{LPPM}(\boldsymbol{p}_0) = \mathbb{Q}] =& Pr[T(\mathrm{LPPM}(\boldsymbol{p}_0)) = t] \cdot h(\mathbb{Q}; t) \\ \leq& (e^{\varepsilon} Pr[T(\mathrm{LPPM}(\boldsymbol{p}_1)) = t] + \delta) \cdot h(\mathbb{Q}; t) \\ \leq& e^{\varepsilon} Pr[\mathrm{LPPM}(\boldsymbol{p}_1) = \mathbb{Q}] + \delta \end{aligned} \tag{21}$$

□

With sufficient statistics, we can prove the privacy of our mechanism in an efficient way. We note that the sample mean of independent Gaussian random variables is a sufficient statistic by Fisher–Neyman factorization theorem. Thus to prove the set of obfuscated outputs satisfies $(r, \varepsilon, \delta, n)$-geo-IND, we only need to prove the mean value of them satisfies $(r, \varepsilon, \delta, 1)$-geo-IND. We first consider the simplest case where the Gaussian mechanism only generates one obfuscated location, and the privacy is satisfied with the following lemma:

Lemma 1. *The* 1*-fold Gaussian mechanism satisfies* $(r, \varepsilon, \delta, 1)$*-geo-IND if*

$$\sigma = \frac{r}{\varepsilon}\sqrt{\ln\frac{1}{\delta^2} + \varepsilon} \tag{22}$$

The proof of Lemma 1 can be found in [13]. Now we consider an n-fold Gaussian mechanism, where we generate n independent samples $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$ from $\mathcal{N}(\boldsymbol{p}, \frac{\sigma^2}{n})$.

Theorem 2. *The n-fold Gaussian mechanism satisfies* $(r, \varepsilon, \delta, n)$*-geo-IND if*

$$\sigma = \frac{\sqrt{n}r}{\varepsilon}\sqrt{\ln\frac{1}{\delta^2} + \varepsilon} \tag{23}$$

Proof. As $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$ are drawn from Gaussian distribution $\mathcal{N}(\boldsymbol{p}, \sigma^2)$, the sample mean $\bar{q}$ distributed as $\mathcal{N}(\boldsymbol{p}, \frac{\sigma^2}{n})$ is a sufficient statistic. According to Theorem 1, we only need $\bar{q}$ to satisfy $(r, \varepsilon, \delta, n)$-geo-IND, which is $\frac{\sigma}{\sqrt{n}} = \frac{r}{\varepsilon}\sqrt{\ln\frac{1}{\delta^2} + \varepsilon}$ by Lemma 1. □

VII. Evaluations and Discussions

A. Experimental Settings

Dataset. To assess the risk of location privacy leakage, we exploit a real-world RTB transaction-log dataset and demonstrate the privacy breaches in existing location-based advertising. We collect 37,262 mobiles users in Shanghai (latitude $\in [30.7, 31.4]$, longitude $\in [121, 122]$), with spatiotemporal data from June 1, 2019 to May 31, 2021, whose size ranges from 20 spatiotemporal points to 11,435 points per user.

Ethical considerations. The real location name, POIs, IMEIs, and the landscapes are hidden from the dataset in the map.

Algorithms. We compare our LPPM algorithm with 2 baseline methods. Our main algorithm refers to the n-fold Gaussian mechanism. The first baseline algorithm is called *naïve post-processing mechanism*, which first obfuscates the real location using 1-fold Gaussian mechanism, and then uniformly samples

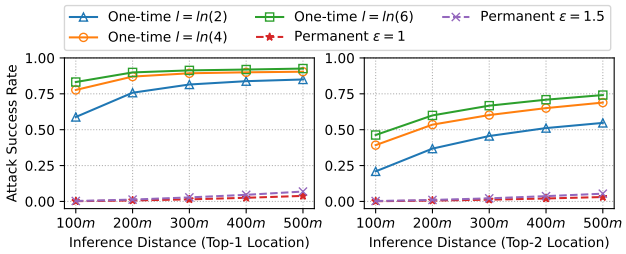


Fig. 6: Performance of the longitudinal attack.

n locations in a certain radius around the obfuscated location. The second baseline algorithm is the *Gaussian mechanism with plain composition*, where we generate n obfuscated locations, each of which satisfies $(r, \frac{\varepsilon}{n}, \frac{\delta}{n}, 1)$-geo-IND to achieve $(r, \varepsilon, \delta, n)$-geo-IND in total by the composition theorem.

Metrics. We measure our system w.r.t the following metrics:
(1) Attack success rate. We say an attack *succeeds* if the inferred top locations are within a threshold distance from the real locations. The *Attack Success Rate* is calculated as the number of users on which our attack succeeds divided by the total number of users.
(2) Utilization rate. The utilization rate is defined in Definition 4. We measure the lower bound of the utilization rate υ given a *confidence level* α, that is

$$Pr(UR \geq \upsilon) = \alpha \tag{24}$$

(3) Efficacy. We randomly generate a set of locations in AOR to measure the probability defined in Definition 5.
(4) Performance. We evaluate the running time of edge devices to provide obfuscation services for multiple users.

Parameter settings. We set $\delta = 0.01$ and $\varepsilon \in \{1, 1.5\}$, which represent a strict and loose privacy level respectively. The indistinguishable radius r is chosen from 500 m, 600 m, 700 m and 800 m. The targeting radius we choose is $R = 5$ km, which is the minimal value of the common interval from 5 km to 25 km as we investigate in the four companies' settings. We choose it because it is more difficult to achieve a better utility for a smaller radius, which can better justify our system. Meanwhile, we set confidence level $\alpha = 0.9$.

Implementation and configuration. To answer the above questions, we implement our algorithms in Scala, and run our system on Raspberry Pi 3. Throughout the experiment, we conduct 100,000 trials for each parameter combination and then use Monte Carlo method to estimate the utility metrics.

B. Evaluating Our Attack and the Effectiveness of Defense

In the one-time geo-IND obfuscation mechanism, we add independent noises to every check-ins to mimic the obfuscated locations used in LBA. We use the original mechanism and set the privacy parameters consistent with the geo-IND paper [9], i.e., $r = 200$ m and $l \in \{\ln(2), \ln(4), \ln(6)\}$, indicating the user can enjoy, resp., $(\frac{\ln(2)}{200}(m^{-1}))$-geo-IND, $(\frac{\ln(4)}{200}(m^{-1}))$-geo-IND, and $(\frac{\ln(6)}{200}(m^{-1}))$-geo-IND. In our permanent obfuscation mechanism, we implement a 10-fold Gaussian mechanism and set the privacy parameters as $r = 500$ m and $\varepsilon \in \{1, 1.5\}$ which is almost the same privacy level as above[3].

Observation-1. One-time geo-IND mechanisms are prone to longitudinal attack, while our permanent obfuscation mechanism can thwart this attack. The attack results are illustrated in Fig. 6. Our results show that the top-1 locations can be accurately recovered under one-time geo-IND mechanisms, with more than 75% of user top-1 locations inferred within 100 m for $l = \ln 4$ and $\ln 6$, and even with the most strict privacy level where $l = \ln 2$, 58% can be inferred within 100 m while 75% within 200 m. The success rates of top-2 locations under one-time geo-IND mechanisms should not be underrated either, since users are still at the risk of privacy breach, with more than 50% possibility to recover top-2 within 200 m when $l = \ln 4$ and $\ln 6$. On the other hand, the attack results also demonstrate our permanent obfuscation mechanism can thwart the longitudinal attack. Less than 1% of user locations can be recovered within 200 m for both top-1 and top-2 locations, and only 6.8% of user top-1 locations and 5% of user top-2 locations can be recovered within 500 m.

C. Algorithm Comparing and Parameter Selection

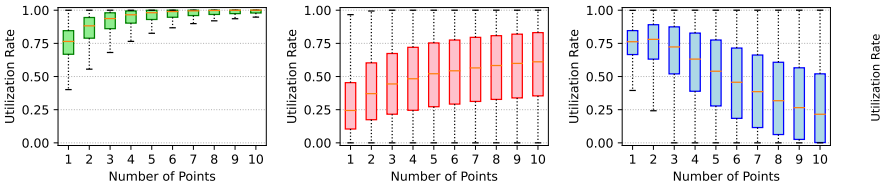
Algorithm comparing. We compare the utilization rate of the n-fold Gaussian mechanism with the baseline mechanisms. The experiment is evaluated with respect to the number n of obfuscated locations ranging from 1 to 10. We fix $\varepsilon = 1$, $r = 500$ m with other default parameters.

Observation-2. Our proposed mechanism outperforms the naïve post-processing mechanism and composition-based Gaussian mechanism, and generating multiple obfuscated locations will decrease the utilization rate in composition-based Gaussian mechanism. As the results show, our n-fold Gaussian mechanism significantly outperforms the other two baseline algorithms. Especially when $n = 10$, our mechanism can achieve almost 100% utilization rate while in naïve post-processing mechanism and composition-based Gaussian mechanism only 58% and merely 20% of utilization rate on average can be achieved respectively. And surprisingly, we find the composition-based Gaussian mechanism fails to enhance the utilization rate by generating multiple obfuscated locations, which means our sufficient statistic provides tighter error bounds on noise composition when generating multiple obfuscated locations, compared to the composition theorem.

Impact of n, ε and r. We evaluate the utilization rate and efficacy with n ranging from 1 to 10, $\varepsilon = 1.0$ and 1.5 and $r = 500m, 600m, 700m, 800m$ respectively. For every parameter combinations, we conduct 100,000 trials for both utilization rate and efficacy.

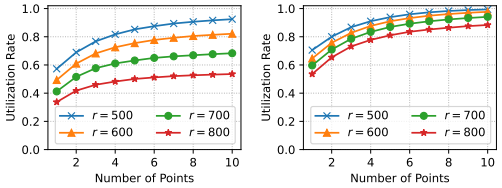
Observation-3. Generating more obfuscated outputs can improve the utilization rate in our n-fold Gaussian mechanism. Fig. 8 shows the minimal utilization rate under different numbers of obfuscated outputs. As the result shows, generating

[3]The paper [9] uses l to denote the privacy level which is equivalent to the ε representation in this paper.



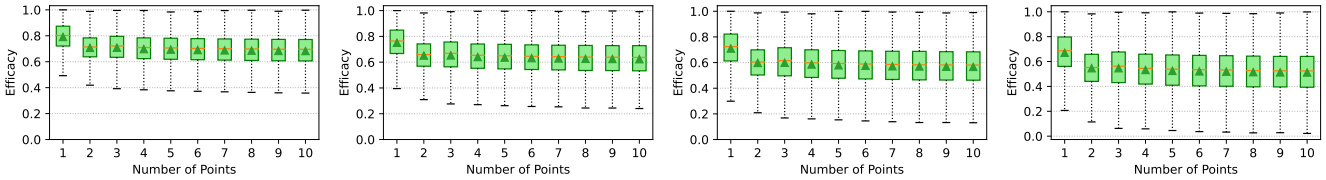
(a) n-fold Gaussian mechanism. (b) Post-processing mechanism. (c) Plain DP composition.

Fig. 7: Utilization rate between different mechanisms, where $\varepsilon = 1$ and $r = 500$.



(a) $\varepsilon = 1$. (b) $\varepsilon = 1.5$.

Fig. 8: Minimal utilization rate where $\alpha = 0.9$.



(a) $r = 500$. (b) $r = 600$. (c) $r = 700$. (d) $r = 800$.

Fig. 9: Efficacy under various r, where $\varepsilon = 1$.

TABLE II: Obfuscation processing time.

Number of Users	2000	4000	8000	16000	32000
Processing Time (s)	340	627	1166	2090	4014

TABLE III: Output selection time.

Number of Users	2000	4000	8000	16000	32000
Processing Time (ms)	90	175	350	698	1377

more obfuscated outputs can improve the utilization rate. When the privacy level is loose ($\varepsilon = 1.5$), our mechanism can improve the utilization rate from 0.6 for $n = 1$ to 0.9 for $n = 10$. Even when the privacy level is strict ($\varepsilon = 1$), the utilization rate can also achieve an improvement by 60% from $n = 1$ to $n = 10$ in general. This indicates our n-fold Gaussian mechanism outperforms the single-fold Gaussian mechanism.

Observation-4. With the output selection module, the efficacy does not significantly decrease with the increase of obfuscated outputs. Fig. 9 illustrates the evaluation of efficacy versus the number n of obfuscated locations, we find the efficacy does not decrease too much compared to generating one candidate output. This means our system will not significantly increase the unwanted advertisements, because the output selection module can select the most useful candidate locations with a higher probability.

D. Scalability of Edge-PrivLocAd

We evaluate the performance of edge devices under multiple users. We use Raspberry Pi 3 to emulate the experiment and evaluate the performance. We first evaluate the time to build a user's location profile and generate candidate locations. We consider our system updates the user's location profile every three months. Our evaluation results are shown in Table II. Since our obfuscation is not a real-time operation, hence the processing time is acceptable in real-world applications. Then we evaluate the performance of the output selection module. Our results in Table III show our system can respond to multiple users with low latency.

VIII. Related Work

This work falls into the intersection of location privacy and mobile advertising security. In this section, we describe related works that are closed to these research areas respectively.

DP for location privacy. Protecting user location privacy with formal and rigorous DP(-like) guarantee has been studied along two directions in the literature. One such line of works has been focusing on releasing user location data while satisfying differential privacy [4]–[8], which is orthogonal to this work. Another direction of prior works explored the query-time protection to user location privacy. Andres et al. proposed *geo-IND* [9], a seminal DP-like framework that perturbs user location through calibrating Laplacian noise. The following work by Bordenabe et al. [20] designed an optimal mechanism based on linear programming, which minimizes the QoS loss while maintaining privacy guarantees. Later works such as [21] and [22] adopted a Bayesian remapping procedure and a multi-step search space pruning approach, resp., to make the optimal process more efficient and practical.

Mobile advertising security. Related works studied the mobile advertising security problem in terms of both *analysis* and *defense*. On the analysis side, some recent work [23]–[26] focused on *measuring impending threats* in mobile advertising, wherein the threat to users' location privacy is also highlighted [24]. On the defense side, many schemes [27]–[31] have been proposed for *private user information gathering* and *private ad delivery*. Most of these schemes incorporated cryptographic approaches like Paillier homomorphic encryption [27], private information retrieval [30], and trusted hardware [28], which make them not efficient enough to respond to an ad request in real-time. The aforementioned mechanisms

are presented for traditional mobile advertising environment, while most recently, Deng et al. [31] proposed a privacy-preserving mechanism to protect user privacy in a real-time bidding (RTB) advertising setting. However, to the best of our knowledge, no similar works except for this paper investigate the location privacy issues in location-based advertising and propose defense mechanisms in this setting.

IX. Conclusion

In this paper, we identify a new type of attack – the longitudinal attack aiming to infer obfuscated locations in LBA scenarios. We propose a de-obfuscation attack to showcase that existing one-time geo-IND based mechanisms can accurately recover the top locations, which poses severe privacy threats to LBA users. To address this privacy issue, we propose a novel edge-assisted system Edge-PrivLocAd to manage the location profile and generate permanent obfuscation for top locations. We prove the privacy of our mechanism using a novel analysis tool, the sufficient statistic which can provide tighter error bounds on noise composition compared to the composition theorem in differential privacy. The experimental results demonstrate the feasibility and effectiveness of Edge-PrivLocAd to provide both rigorous privacy guarantee and robust utility.

Acknowledgment

This research was supported in part by National Natural Science Foundation of China under Grants No. 62132013, 72171145, and 61972453.

References

[1] H. Yu, E. Wei, and R. A. Berry, "Analyzing location-based advertising for vehicle service providers using effective resistances," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, no. 1, pp. 1–35, 2019.

[2] P. Cheng, X. Lian, L. Chen, and S. Liu, "Maximizing the utility in location-based mobile advertising," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 776–788, 2020.

[3] "Location based advertising (lba) - global market trajectory & analytics," Global Industry Analysts, Inc, Tech. Rep., 2021.

[4] R. Chen, G. Acs, and C. Castelluccia, "Differentially private sequential data publication via variable-length n-grams," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, 2012, pp. 638–649.

[5] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1298–1309.

[6] X. He, G. Cormode, A. Machanavajjhala, C. M. Procopiuc, and D. Srivastava, "Dpt: differentially private trajectory synthesis using hierarchical reference systems," *Proceedings of the VLDB Endowment*, vol. 8, no. 11, pp. 1154–1165, 2015.

[7] X. He, N. Raval, and A. Machanavajjhala, "A demonstration of visdpt: Visual exploration of differentially private trajectories," *Proceedings of the VLDB Endowment*, vol. 9, no. 13, pp. 1489–1492, 2016.

[8] M. E. Gursoy, L. Liu, S. Truex, L. Yu, and W. Wei, "Utility-aware synthesis of differentially private and attack-resilient location traces," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 196–211.

[9] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, 2013, pp. 901–914.

[10] K. Fawaz and K. G. Shin, "Location privacy protection for smartphone users," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 239–250.

[11] K. Fawaz, H. Feng, and K. G. Shin, "Anatomization and protection of mobile apps' location privacy threats," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 753–768.

[12] W. Eltarjaman, R. Dewri, and R. Thurimella, "Location privacy for rank-based geo-query systems," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 77–96, 2017.

[13] L. Zhou, L. Yu, S. Du, H. Zhu, and C. Chen, "Achieving differentially private location privacy in edge-assistant connected vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4472–4481, 2018.

[14] S. Sun, L. Yu, X. Zhang, M. Xue, R. Zhou, H. Zhu, S. Hao, and X. Lin, "Understanding and detecting mobile ad fraud through the lens of invalid traffic," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 287–303.

[15] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[16] S. Yuan, J. Wang, and X. Zhao, "Real-time bidding for online advertising: measurement and analysis," in *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, 2013.

[17] Google Ads: grow your business with google ads. https://ads.google.com/intl/en_US/home/. Accessed: 2022-01-27.

[18] C. Qiu, A. C. Squicciarini, C. Pang, N. Wang, and B. Wu, "Location privacy protection in vehicle-based spatial crowdsourcing via geo-indistinguishability," *IEEE Transactions on Mobile Computing*, 2020.

[19] Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang, and H. Qi, "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1330–1341, 2019.

[20] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 251–262.

[21] K. Chatzikokolakis, E. Elsalamouny, and C. Palamidessi, "Efficient utility improvement for location privacy," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 308–328, 2017.

[22] R. Ahuja, G. Ghinita, and C. Shahabi, "A utility-preserving and scalable technique for protecting location data with geo-indistinguishability," in *Advances in Database Technology - 22nd International Conference on Extending Database Technology*, 2019, pp. 217–228.

[23] A. Razaghpanah, R. Nithyanand, N. Vallina-Rodriguez, S. Sundaresan, M. Allman, C. Kreibich, and P. Gill, "Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem," in *25th Annual Network and Distributed System Security Symposium*, 2018.

[24] G. Venkatadri, A. Andreou, Y. Liu, A. Mislove, K. P. Gummadi, P. Loiseau, and O. Goga, "Privacy risks with facebook's pii-based targeting: Auditing a data broker's advertising interface," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 89–107.

[25] G. Chen, W. Meng, and J. Copeland, "Revisiting mobile advertising threats with madlife," in *The World Wide Web Conference*, 2019, pp. 207–217.

[26] M. Zhang, W. Meng, S. Lee, B. Lee, and X. Xing, "All your clicks belong to me: investigating click interception on the web," in *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 941–957.

[27] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas, "Adnostic: Privacy preserving targeted advertising," in *Proceedings Network and Distributed System Symposium*, 2010.

[28] M. Backes, A. Kate, M. Maffei, and K. Pecina, "Obliviad: Provably secure and practical online behavioral advertising," in *2012 IEEE Symposium on Security and Privacy*, 2012, pp. 257–271.

[29] M. Hardt and S. Nath, "Privacy-aware personalization for mobile advertising," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, 2012, pp. 662–673.

[30] M. Green, W. Ladd, and I. Miers, "A protocol for privately reporting ad impressions at scale," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1591–1601.

[31] E. Deng, H. Zhang, P. Wu, F. Guo, Z. Liu, H. Zhu, and Z. Cao, "Pri-rtb: Privacy-preserving real-time bidding for securing mobile advertisement in ubiquitous computing," *Information Sciences*, vol. 504, pp. 354–371, 2019.