Überprivate: Differentially Private Ride-Sharing

Ashley Chen, Amelia Meles, Nicholas Ramirez

May 4, 2025

1 Introduction

Popular ridesharing apps like Uber and Lyft collect a lot of personal information and location data on riders. Some of this information is viewable by rideshare drivers as well. For example, until 2018, the Uber app allowed drivers to view the precise pickup and dropoff locations for all of their past riders. While there has been a recent effort to reduce data collection and limit sharing with drivers, the minimum data these apps claim to be necessary still pose a privacy risk to users since they include dropoff address, pickup address, and first name at minimum.

We propose a novel design for a ridesharing app, Überprivate, that protects privacy using the strict mathematical guarantees of Differential Privacy (DP). We use a relaxed DP model where the central ride-sharing company is treated as a trusted curator of the user's location data and the driver is untrusted. In this model we use differentially private mechanisms specific to location data, called Geo-Indistinguishability, to perturb the user's locations for pickup and dropoff according to their desired privacy level for each location and their desired walking radius. Although using DP for enhancing ridesharing and carpooling privacy has been studied, we propose a novel solution for the under-studied security model that provides privacy from drivers. Furthermore, our solution prioritizes customizable and understandable privacy for the user. We believe that by making a relaxation to DP such that each driver has a fresh privacy budget for each user, we can achieve our privacy and utility goals.

2 Background

2.1 Ride-Sharing and Privacy Risks

There are several privacy risks that motivate the threat models that are formalized in the Project Scope section below. First, a primary concern is the data that the ridesharing application collects on every single ride; this data could be shared and used in a way that harms the user. As a severe example, a ridesharing app may deduce from various dropoffs at a cigar bar that a user is a smoker; if they decide to share that information with the user's insurance company, their rate could skyrocket. Second, safety is a primary concern for users of a ridesharing app, and safety may be threatened by any driver that can learn their habits and locations of interest. So, the first risk derives primarily from the central ridesharing app while the second risk derives from the drivers.

2.2 Potential Threat Models

Our project identifies two threat models in this ride-sharing use-case, though we focus on one that is less studied in the literature. In the first model, we identify the case where there are malicious passive drivers and a trusted centralized company. Here, we choose to exclude active attackers that could change or falsify rider data because that would interfere with the correctness of the rideshare app, which is a self-interest that rideshare companies would already be invested in protecting. The drivers act as passive adversaries by using the sensitive personal information they learned during the rides to seek out riders on their personal time.

In the second threat model, we consider the case where the rideshare app uses self-driving cars. As a result, the sole adversary arises from the existence of a self-interested rideshare company who may collect and store of vast amounts of personal data for potentially malicious use. Although this threat model is also a passive attack and is very similar to the first in its agents, it ignores the privacy risks presented by the driver.

As will be discussed in the related work section, most prior work focuses on models similar to that of the second one. As such, we focus on the first threat model, where we have malicious passive drivers and a trusted centralized company.

2.3 Differential Privacy

We are motivated to apply differential privacy to solve this problem because of its strong mathematical notions of privacy. In differential privacy, a data practitioner can control the strength of the privacy guarantee by tuning the privacy parameter ϵ, also called a privacy loss or privacy budget. The lower the value of the ϵ parameter, the more indistinguishable the results, and therefore the more each individual's data is protected. The application of noise to the results of the queries mathematically bounds the risk of an adversary learning about any given individual's presence in the dataset. One of the most common noise mechanisms in DP is random sampling from the Laplace distribution. However, making the data more private by adding noise inherently means making the data less useful.

The key tradeoff when using Differential Privacy is between privacy and utility. Currently, users in ridesharing choose their pickup and dropoff, so they could intentionally obfuscate their true travel plans in theory, but in practice few people do this. However, if this process was automated and customized to suit the user's desired level of privacy, some users may be willing to lose some utility (i.e. walking farther to their pickup) to gain some privacy and therefore peace of mind.

2.4 Geo-Indistinguishability

Geo-Indistinguishability is an interpretation of differential privacy with a location focused lens that was introduced by Andres, et al. [1]. It inherits useful properties from the differential privacy framework, such as composability of the privacy loss budget across queries. Similar to DP, it bounds the adversary's ability to learn information about someone's true location based on a noisy

location. However, while classic DP concerns the ability of an adversary to distinguish between two databases (one with and one without a given record), geo-indistinguishability concerns the ability of an adversary to distinguish between two locations, x and x', based on euclidean distance. Similarly, as classic DP uses noise mechanisms that sample random noise from a distribution like the Laplace distribution, geo-indistinguishability uses the Planar Laplace mechanism, which samples a random θ and a random r that is scaled by the privacy parameter ϵ according to the planar Laplace distribution to produce a new, noisy location. To improve utility, the authors suggest extra optimization steps:

1. They define an "Area of Interest" which is the area around the user's true location x that they consider relevant (i.e. our walking radius)
2. They calculate the noisy location x' as described above, and use that for the location based query (i.e. finding nearby points of interest) with a larger "Area of Retrieval" radius. This radius doesn't impact privacy but has potential impacts on accuracy of the mechanism as well as its speed and network overhead.
3. They filter the results from the "Area of Retrieval" to use only those within the user's "Area of Interest"

Classically, geo-indistinguishability is based purely on privacy loss parameter ϵ and euclidean distance, and treats the planar coordinate space uniformly. But, in reality, our maps have a lot of semantic information about locations that is lost by treating the entire space the same. Therefore, Chatzikokolakis et al. explored the use of "elastics metrics" that factor population density and other location semantics into the Geo-Indistinguishability noise mechanism [2].

3 Related Work

In our project, we will utilize Differential Privacy as defined in the extensive work by Cynthia Dwork [3]. This provides a rigorous mathematical definition of privacy which enables us to quantify privacy in the context of ridesharing.

Beyond definitions, Hesselmann, Gertheiss, and Müller provide insight into the current state of practice for handling user data in popular ridesharing apps [4]. They provide nicely merged attributes such as contact information, social media, personal description, interests, job, and address (each of these merged attributes have smaller atomic attributes), and the general protection of these attributes among 12 ridesharing companies.

Fatima Errounda and Yan Liu provide a detailed survey involving the many approaches to differential privacy on data that involves locations and trajectories [5]. Specifically, they introduce and detail general approaches to location privacy such as distance-based, obfuscation-based, and anonymity-based methods. Distance-based methods focus on shifting the indistinguishability to the observed locations of a single user. Obfuscation-based is focused on sharing an approximation of the true location without the need for it to correlate with the true location. Lastly, anonymity-based methods focus on making partitioned spaces (that consist of users' locations) indistinguishable, such as in [6] and [7].

The general approaches above correspond to location-based masking. It is important to note that most of the current literature involves calculating an

optimal meeting-point, masking the true location (k-anonymity, obfuscation, etc.), and using differential privacy to achieve "geo-indistinguishability". Another approach adopted by Uber is elastic sensitivity [8]. This approach focuses on enforcing differential privacy to Uber databases by calculating the sensitivity of a query without having to make changes to the database itself.

The total range of the related work we have presented reveals how complex protecting user data in ridesharing can be. This complex lies in the many problems that need to be considered such as the scheduling of the rides, optimization of the total path (pickup, dropoff, and general path), and overall service quality of the given app. As a result, most of the academic papers focus on the above complexes. Our project will take on a strong privacy perspective instead of focusing on the numerous other potential considerations in ridesharing. Moreover, we consider a unique threat model that is not considered in other attempts at privatizing ride-sharing, because most prior work focuses on protecting user data from the central application rather than from the drivers.

4 Proposed Solution

4.1 Goals

As discussed in prior sections, we prioritize protecting the user in a threat model that contains potentially malicious drivers while providing an understandable interface for users to customize and understand their privacy with. As such we define the following goals for our proposed solution:

- Protect User Locations from Drivers
- Enable User-Controlled Privacy Levels on a Per-Location basis
- Enable User-Controlled Walking Radii on a Per-Location basis
- Incorporate Context-Awareness of Location Semantics into the Noise Mechanism

We outlined in prior sections why location privacy from drivers is important. For user-controlled privacy parameters, we consider this an important goal because users want to keep different locations more private than others (i.e. home versus work office), and they might be willing to walk more or less based on that location and their desired privacy level. Moreover, we think location semantics, like population density, are important factors that can contribute to the noise level added, because, for example, dense urban areas give you a much stronger ability to "hide amongst the crowd" as compared to rural areas.

4.2 System Overview

Überprivate is our proposed solution to the described problem and threat model. The system we propose uses a noise mechanism that satisfies geo-indistinguishability by perturbing the user's pickup and dropoff location according to their per-location privacy level and walking radii settings. This noise mechanism uses an elastic metric, as proposed in [2], that factors population density of the map into the noise mechanism to provide more a realistic tradeoff between privacy

and utility. The driver will then pick up and drop off the user at the noisy locations shared by the ride-sharing app. Although the driver matching algorithm is mostly a black box for this use case, we specify that it must incorporate budget tracking such that each driver can only pick up the same user for a given location a bounded number of times according to the user-specified privacy loss budget, which will be further detailed below.

4.3 Helping Users Understand Privacy Parameters

Figure 1: Location Configuration User Interface

The first step for a user in our system when taking a ride from a new place is to specify the desired privacy parameters for that location. In geo-indistinguishability, the privacy loss budget for a certain location is specified as $\epsilon = \frac{l}{r}$ where l is the desired privacy level for that location and where r is the radius in which you want to protect that location to that level. In Figure 1 we present mock-ups of how we would provide. Notably, they can use a sliding scale to determine their radius of pickup. This value corresponds to the "Area of Interest" in the generalized geo-indistinguishability literature, . Though this "Area of Interest" radius can actually be a different value from r, we set them equal for simplicity in this system so the user has fewer free parameters to concern themselves with. Then, the user would similarly choose their privacy level

for the location in a sliding scale user interface as well. Lastly, the interface would show the user their privacy protection for that location based on their configurations as a binned value: "low", "medium", or "high". This binning is a crucial simplification step that frames privacy in a less precise, but much more understandable way.

These configurations are crucial for the user experience. As discussed, a user might really value their location privacy for their home, but not for a very public place like the grocery store or airport. Similarly, they may be willing to walk farther in order to get this stronger privacy for certain locations, or they might not be willing to walk far, in which case it might severely limit how many times a given driver can pick them up around a sensitive location.

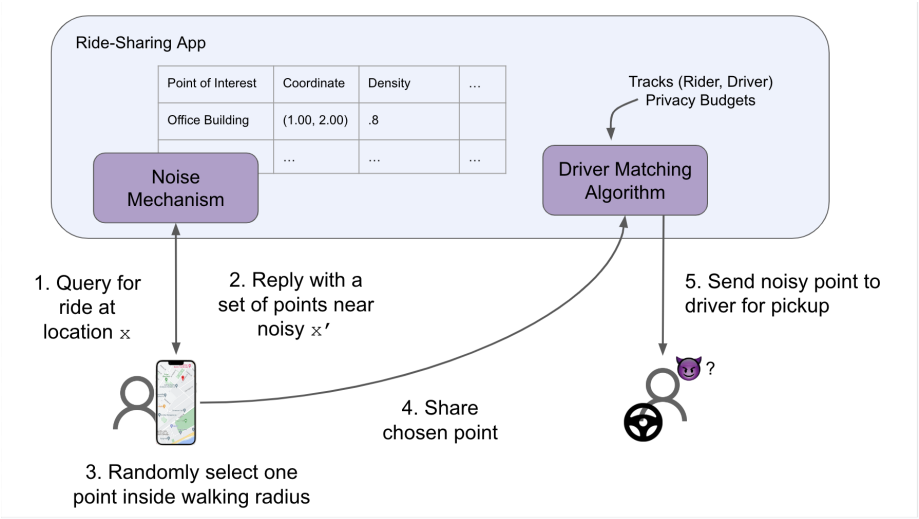

Figure 2: Diagram of the Ride Querying Process for a User

4.4 Ride Querying System

Now we further elaborate on our solution for the Ride Querying system, which is summarized in Figure 2. We follow the optimized model for geo-indistinguishable applications with high utility needs as described in Section 2.4. First, the user sends their location privacy configurations to the ride-sharing app as described in above, which sets up the total privacy budget ϵ that each driver would be able to expend for picking them up at a given location x. Then, they query for a ride at their desired location x. Überprivate applies noise to x to produce a ϵ_0-geo-indistinguishable location x', where ϵ_0 is some fraction of ϵ that would need to be decided by the ride-sharing app experimentally to balance the utility (i.e. reducing how far a user needs to walk) with the limit on the number of times a given driver can pick up a user in that place[1]. The noise mechanism uses the "elastic metric" for distance as proposed in [2] that essentially warps the

[1]This may be important, for instance, if a given driver can only pick them up at a certain location once; eventually, a user will likely face higher wait times as the app would need to find a driver with a non-expended budget, which will become more and more difficult as this population shrinks.

geometrical distance measurement to incorporate factors of location semantics like population density.

This noise mechanism returns noisy x', which Überprivate uses to generate a set of points of interest (or just coordinates on the plane) in the area surrounding x'. Then as described in the optimized technqiue in 2.4, the user will filter the results to only include those withing the walking radius, and then it will choose a random point amongst these to select as the pickup location. This noisy point-of-interest is inputted into the Driver Matching Algorithm. The main invariant we care about in this algorithm is that no driver ever expends their budget for a given user and location. Though this might mean that a user may eventually have no eligible driver matches, we believe this can be configured to be very, very unlikely. In the last step, the noisy point-of-interest is sent to the driver, who should not be able to learn much about the user's true sensitive location based on the pickup location.

4.5 Budget Relaxation

Crucially, we make the relaxation that for each user's location, each driver has their own budget to expend. In classical DP, a privatized dataset release is assumed to be known by everyone once released, so their is no "per-person" privacy budget for each person who sees the query result. Rather, there is a global tracking of privacy loss incurred by each release. As such, if we used this classic model in our system, our privacy loss would grow in an unbounded fashion, since every driver who picked someone up from a given location would erode the global privacy loss further. Therefore, we consider a more realistic model where non-colluding drivers are assumed to keep information to themselves and have separate privacy loss budget tracking from one another.

5 Discussion

5.1 Future Work

Though we originally started with the goal of protecting the user from both the central ride-sharing app and the drivers, it quickly appeared to us as a much more difficult problem. However, we think future work in this area could once again attempt this. For instance, we can imagine an alternate system where the geo-indistinguishability noise mechanism is applied locally on a user's phone, though the elastic metric model of noise might be too large and computationally heavy for some devices. In theory, this would ensure that even the central app never sees the true location.

If we did not pursue this more difficult threat model, we could pursue other optimizations of Überprivate instead. For example, since the central app is trusted, it could potentially cache the noisy locations for each user's sensitive location, and reuse the value without further expending the privacy budget for a certain driver. This would allow a driver to pick up a given rider in the same noisy x' more than once without further privacy loss as a form of "post-processing", which differently privacy is immune to [3]. This seems like a promising idea but would need further security analysis to ensure there is no form of privacy leakage.

5.2 Conclusion

The aim of this project was to highlight how user misconceptions leave them vulnerable to privacy risks in ride-sharing contexts and how we might incorporate ideas of differential privacy to reduce privacy loss and aid user understanding. Some users greatly value their location privacy, but configurability in the level of privacy is important given how widely user attitudes vary on sensitivities of different locations. Overall, Überprivate incorporated our goals of user-controlled location privacy levels and walking radii to protect user locations from drivers to their specified level. We incorporate the idea of elastic distance metrics to account for context-awareness of location semantics in the application of noise, which allows for more realistic privacy protection rather than a uniform treatment of space. However, to prevent unbounded privacy loss, we make the relaxation that each user's location has a privacy loss budget that can be used independently in parallel by different drivers, rather than a global budget that would become quickly expended.

References

[1] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 901–914, 2013.

[2] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. Constructing elastic distinguishability metrics for location privacy. *Proceedings on Privacy Enhancing Technologies*, 2015(2):156–170, jun 2015.

[3] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis., 2006.

[4] Carsten Hesselmann, Jan Gertheiss, and Jörg P Müller. Ride sharing & data privacy: An analysis of the state of practice. *arXiv preprint arXiv:2110.09188*, 2021.

[5] Fatima Zahra Errounda and Yan Liu. An analysis of differential privacy research in location and trajectory data. 2020.

[6] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. Protection of location privacy using dummies for location-based services. In *21st International conference on data engineering workshops (ICDEW'05)*, pages 1248–1248. IEEE, 2005.

[7] Mingqiang Xue, Panos Kalnis, and Hung Keng Pung. Location diversity: Enhanced privacy protection in location based services. In *Location and Context Awareness: 4th International Symposium, LoCA 2009 Tokyo, Japan, May 7-8, 2009 Proceedings 4*, pages 70–87. Springer, 2009.

[8] Noah Johnson, Joseph P Near, and Dawn Song. Towards practical differential privacy for sql queries. *Proceedings of the VLDB Endowment*, 11(5):526–539, 2018.