Privacy-Aware Mobile Services over Road Networks

Ting Wang
College of Computing
Georgia Institute of Technology
twang@cc.gatech.edu

Ling Liu
College of Computing
Georgia Institute of Technology
lingliu@cc.gatech.edu

ABSTRACT

Consider a mobile client who travels over roads and wishes to receive location-based services (LBS) from untrusted service providers. How might the user obtain such services without exposing her private position information? Meanwhile, how could the privacy protection mechanism incur no disincentive, e.g., excessive computation or communication cost, for any service provider or mobile user to participate in such a scheme? We detail this problem and present a general model for privacy-aware mobile services. A series of key features distinguish our solution from existing ones: a) it adopts the *network-constrained* mobility model (instead of the conventional *random-waypoint* model) to capture the privacy vulnerability of mobile users; b) it regards the attack resilience (for mobile users) and the query-processing cost (for service providers) as two critical measures for designing location privatization solutions, and provides corresponding analytical models; c) it proposes a robust and scalable location anonymization model, XStar, which best leverages the two measures; d) it introduces multi-folded optimizations in implementing XStar, which lead to further performance improvement. A comprehensive experimental evaluation is conducted to validate the analytical models and the efficacy of XStar.

1. INTRODUCTION

With ubiquitous wireless connectivity and continued advance in mobile positioning technologies (e.g., GPS-equipped devices, cellular phones), recent years have witnessed the explosive growth of location-based services (LBS). Examples include location-based store finders ("*Where is the nearest gas station to my current location?*"), traffic condition tracking ("*What is the traffic condition on Highway 85 North?*"), and spatial alarms ("*Remind me to drop off a letter when I am near a post office.*"). Mobile clients obtain such services by issuing queries together with their location information to the LBS providers. While offering everyday convenience and business opportunities, LBS also opens the door for potential misuse of mobile users' private location information [17, 29]. For example, the collected location information can be exploited to spam users with unwanted advertisements, execute physical stalking [7, 27], or perform inference about personal medical records by knowing users' frequent visits to specific clinics.

A plethora of work has been done on the anonymization of location information for mobile users [1, 3, 6, 8, 9, 11, 15, 16, 24, 30]. Nevertheless, assuming the *random-waypoint* mobility model [4, 14], wherein users can move in arbitrary directions at random speed, most existing solutions fail to address the vulnerabilities of mobile users traveling over roads, where both the user mobility and the location-based service processing are constrained by the underlying road networks.

More specifically, the protection enough under the random-waypoint model might be insufficient under the *network-constrained* mobility model. For example, the spatial cloaking techniques [1, 8, 9, 11, 24, 30] protect users' privacy by blurring their exact positions with cloaked spatial areas, and measure the amount of protection as the area size. Such a metric, however, is inapplicable under the road network model, for a large area might contain a single road segment, which enables the adversary to track down the mobile user fairly easily. Furthermore, the condition of the road network, e.g., the network topology, has significant impact over the query-evaluation and communication efficiency, which should be a critical concern for developing location privatization solutions. For instance, the complexity of computing the network distance of two points, a most fundamental operation in location-based query processing, varies considerably with the underlying network structure.

In this work, we present a general framework for location privacy protection under the network-constrained mobility model. Compared with prior work, our framework highlights three distinct features.

First, we argue that the protection for mobile users' privacy should be provided along two orthogonal dimensions: (1) *location anonymity*, which advocates that it should be difficult to identify a specific user among a set of users, an *anonymous set*, based on their location information; and (2) *location diversity*, which promotes that it should be difficult to link a specific user with a specific location (such as a road segment) with high certainty. Furthermore, such privacy requirements should be customizable and supported on a per query basis. In this following, we refer to the process of achieving location anonymity and diversity as *location anonymization*.

VLDB '09, August 24-28, 2009, Lyon, France

Second, we regard the attack resilience of the performed protection and the processing cost of the query with anonymous location information (including both computation and communication costs) as two critical measures for designing location privatization solutions. We propose corresponding analytical models. In particular, we reveal the inherent trade-off between these two metrics through a formal study of two basic anonymization models.

Third and most importantly, we present XStar, a novel star-graph-based location anonymization model, to achieve the optimal balance between high query-processing efficiency and robust inference-attack resilience. To the best of our knowledge, this is the first model taking account of both measures. In implementing XStar, we introduce a suite of optimization strategies to further enhance its performance. Extensive experimental evaluation is conducted to validate the analytical models and the efficacy of XStar.

The remainder of the paper will be organized as follows. In Section 2, we introduce fundamental concepts and models, and discuss the design objective of location privatization solutions. Section 3 describes in detail the design of XStar. The theoretical analysis of XStar in terms of query-processing cost and inference-attack resilience is presented in Section 4 and Section 5. Section 6 addresses detailed issues of implementing XStar and proposes multi-folded optimization strategies. The proposed solution is empirically evaluated in Section 7. Section 8 surveys relevant literatures. The paper is concluded in Section 9.

2. CONCEPTS AND MODELS

In this section, we start with introducing the concept of road-network-aware location privacy, and then present the model of anonymous query processing; through an analysis of two basic anonymization models in terms of inference-attack resilience and query-processing cost, we outline the design objective of XStar.

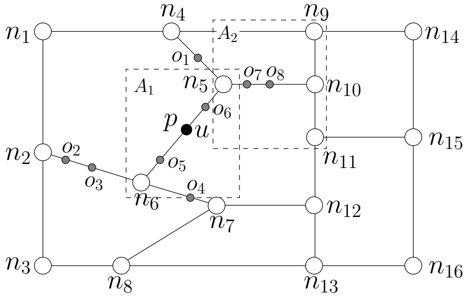


Figure 1: A road network model.

2.1 Road Network Model

In this paper, we model a road network as an un-directed graph $G = (\mathcal{V}_G, \mathcal{E}_G)$, with the node set $\mathcal{V}_G$ and the edge set $\mathcal{E}_G$ representing road junctions and direct road links, respectively. An example of the road network model is shown in Figure 1. We use $\mathsf{d}_G(n)$ to denote the degree of a node n with respect to the graph G. Specifically, n is called an *intersection* node if $\mathsf{d}_G(n) \geq 3$, an *intermediate* node if $\mathsf{d}_G(n) = 2$, and an *end* node if $\mathsf{d}_G(n) = 1$.

To model the restriction of users' mobility by the underlying road network, we introduce the concept of *segment*: a segment s is a sequence of edges $(\overline{n_0 n_1}, \overline{n_1 n_2}, \ldots, \overline{n_{L-1} n_L})$[1] where $\{n_i\}_{i=0}^{L}$ are all distinct, and the degrees of the nodes satisfy $\mathsf{d}_G(n_i) \geq 3$ for $i = 0$ or L, and $\mathsf{d}_G(n_i) = 2$ otherwise. That is, n_0 and n_L are intersection or end nodes, and all others are intermediate nodes.

Note that each edge is either a segment itself, or belongs to a unique segment; that is, a road network can be uniquely partitioned into a set of segments. We therefore assume the following scenario: every mobile user registered with LBS is moving along certain road segment, and sends her location-based query together with her current position information to the LBS provider, which then executes the query based on the provided location information.

2.2 Location Privacy Model

We consider two types of privacy concerns arising in LBS under the network-constrained mobility model, namely *location anonymity* and *location diversity*.

The first requirement ensures the indistinguishability of a specific mobile user among a set of users (an *anonymous set*), and is usually captured by the concept of *location k-anonymity* [8, 11].

Definition 1 (Location k-anonymity). *A user's reported location is said to be k-anonymous, if at least $(k-1)$ other active users report the same location.*

Ensuring location anonymity alone (the objective of most prior work [1, 8, 9, 11, 24]), however, does not provide sufficient protection when the underlying road network is taken into consideration. For example, in Figure 1, assume that users $\mathtt{u}_1$ and $\mathtt{u}_2$ publish their k-anonymous location as A_1 and A_2, respectively. Given that A_1 and A_2 are of equal size and contain identical number of active users, $\mathtt{u}_1$ and $\mathtt{u}_2$ are considered to enjoy equivalent amount of privacy protection, under the criterion of location anonymity; however, it is much easier for the adversary to track down $\mathtt{u}_1$ than $\mathtt{u}_2$, since $\mathtt{u}_1$ is associated with a single road segment $\overline{n_5 n_6}$, while $\mathtt{u}_2$ is possibly associated with three $\{\overline{n_5 n_{10}}, \overline{n_9 n_{10}}, \overline{n_{10} n_{11}}\}$. Intuitively, from the adversary's perspetive, the difficulty of tracking a user is in proportion to the number of segments that she is possibly associated with. This motivates us to introduce location diversity [19] as the second dimension of privacy measure.

Definition 2 (Segment l-diversity). *A user's published location is said to be l-diverse, if it satisfies location k-anonymity, and contains at least l different road segments.*

In our framework, every mobile user $\mathtt{u}$ specifies customized privacy requirement as $(\delta_k^{\mathtt{u}}, \delta_l^{\mathtt{u}})$ in terms of k-anonymity and l-diversity on a per query basis. Besides, to guarantee the quality of received services, e.g., response time, $\mathtt{u}$ may also specify customized QoS requirement as maximum spatial tolerance $\sigma_s^{\mathtt{u}}$ and temporal tolerance $\sigma_t^{\mathtt{u}}$; the former bounds the expansion of the anonymous location, while the latter specifies the tolerable delay due to the anonymization operation (if a request could not be honored within $\sigma_t^{\mathtt{u}}$, it is typically discarded). This set of parameters $(\delta_k^{\mathtt{u}}, \delta_l^{\mathtt{u}}, \sigma_s^{\mathtt{u}}, \sigma_t^{\mathtt{u}})$ is called $\mathtt{u}$'s *service profile*.

To fulfill such requirements, we introduce the operation of *location anonymization*.

[1] Without ambiguity, we use a sequence of nodes $\overline{n_0 \ldots n_L}$ to denote the segment $(\overline{n_0 n_1}, \overline{n_1 n_2}, \ldots, \overline{n_{L-1} n_L})$.

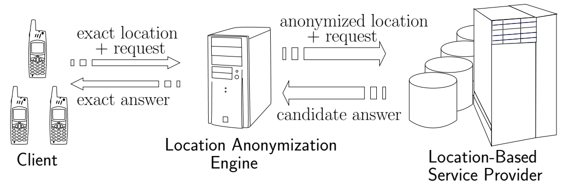


Figure 2: A framework of privacy-aware mobile service.

Definition 3 (Location Anonymization). *Let q denote a location-based query issued by a mobile user u. Location anonymization transforms the exact location information associated with q to an approximate version that satisfies u's service profile.*

With road networks as the background context, we assume that the anonymization operation is performed on the basis of road segment, and an anonymous location is composed by a set of segments. It is noted that for ease of exposition, the factor of segment length is not discussed in this paper; handling the heterogeneity of segment lengths, however, is fairly straightforward, e.g., specify high diversity requirement for short segments, or logically divide a long segment into a set of short ones.

Also, we assume a trusted, third-party *location anonymization engine* (LAE) that acts as a middle layer between mobile users and LBS providers, and performs location anonymization. It is noted that compared with alternatives, this centralized LAE architecture demonstrates a set of critical advantages. (1) It is much easier to provide security protection and operation regulation for a single LAE than a huge number of individual LBS providers with conflicting commercial interests. (2) With the help of LAE, one can achieve privacy guarantees that are unachievable by client-based or peer-to-peer architectures (e.g., [9, 30]), such as location anonymity and diversity. (3) Moreover, this architecture has been successfully applied in a variety of location privatization systems [1, 2, 3, 8, 11, 24].

Specifically, LAE is responsible for (1) receiving the query and the exact position information from the mobile user; (2) anonymizing the location information according to the user's privacy requirements, and relaying it to the LBS provider; (3) extracting the exact query result from the candidate answer returned by the service provider by properly filtering false positive information (a detailed discussion in [1]); and (4) delivering the exact answer to the client. Figure 2 sketches this framework.

2.3 Anonymous Query Processing Model

Now, we consider the processing of queries with anonymous location information (a set of segments). Without loss of generality, we focus our discussion on k nearest neighbors (k-NN) style queries, with which the user requests for the k *objects of interest* with the minimum distances to her current position, the *query focal point*. The distance between two points on the road network is defined as the length of their shortest path. The extension to other query types, e.g., range queries, is referred to our technical report [28].

Intensive research has been directed to the query processing for spatial network databases recently [5, 13, 20, 25, 26]. While the proposed approaches differ in assumptions and techniques, we abstract two fundamental operations underlying these approaches to construct our model: (1) segment-based operation, which takes the query q and a segment s as input, and returns the set of objects on s that satisfy the query predicate, denoted by $\mathcal{O}(\mathsf{q}, s)$; and (2) node-based operation, which takes q and a node n as input, and returns the set of objects in the vicinity of n that satisfy the query predicate, denoted by $\mathcal{O}(\mathsf{q}, n)$.

We base our basic query processing model on the following theorem (the proof is omitted due to the space constraint).

Theorem 1. *For a k-NN query q with query focal point p on a segment s, with n_0^s and n_1^s as the two ends of s, the query result $\mathcal{R}(\mathsf{q}, p)$ satisfies the following condition:*

$$\mathcal{R}(\mathsf{q}, p) \subseteq \mathcal{O}(\mathsf{q}, s) \cup \mathcal{O}(\mathsf{q}, n_0^s) \cup \mathcal{O}(\mathsf{q}, n_1^s)$$

This theorem amounts to saying that the result of q must be included in the union of the following two sets of objects: (1) the objects of interest on s, and (2) the k nearest objects of interest to n_0^s and n_1^s. An example is given in Figure 1. A user u issues a k-NN query q with $k = 3$ when moving on the segment $\overline{n_5 n_6}$. It is clear that the exact answer $\mathcal{R}(\mathsf{q}, p)$ $= \{o_5, o_6, o_7\}$ appears in the union of $O(\mathsf{q}, \overline{n_5 n_6}) = \{o_5, o_6\}$, $\mathcal{O}(\mathsf{q}, n_5) = \{o_1, o_6, o_7\}$, and $\mathcal{O}(\mathsf{q}, n_6) = \{o_5, o_3, o_4\}$.

Hence, given a query q with its focal point on a segment s, the processing of q comprises one segment-based operation with respect to s and two node-based operations with respect to n_0^s and n_1^s. We now extend this model to the case of anonymous queries involving multiple segments. We first introduce the concept of *boundary node*.

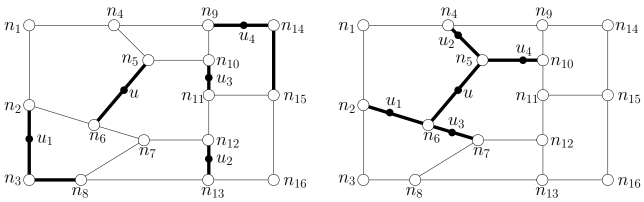
Definition 4 (Boundary Node). *Given a set of segments S in the road network G, the set of boundary nodes of S, denoted by $\mathcal{BV}_S$, is defined as:*

$$\mathcal{BV}_S = \{n | n \in \mathcal{V}_S, \mathsf{d}_G(n) > \mathsf{d}_S(n)\}$$

That is, $\mathcal{BV}_S$ are those nodes in S that are connected to the rest of the network. For instance, for the set of segments S $= \{\overline{n_2 n_1 n_4}, \overline{n_2 n_6}, \overline{n_2 n_3 n_8}\}$ in Figure 1, its boundary node set is given as $\mathcal{BV}_S = \{n_4, n_6, n_8\}$.

For a query q with associated anonymous location as a set of segments S, the evaluation of q consists of two parts: (1) the objects of interest on the segments of S, i.e., $\cup_{s \in S} \mathcal{O}(\mathsf{q}, s)$; and (2) the results as q issued on the boundary nodes of S, i.e., $\cup_{n \in \mathcal{BV}_S} \mathcal{O}(\mathsf{q}, n)$. Formally,

$$\mathcal{R}(\mathsf{q}, S) \subseteq (\cup_{s \in S} \mathcal{O}(\mathsf{q}, s)) \cup (\cup_{n \in \mathcal{BV}_S} \mathcal{O}(\mathsf{q}, n))$$



(a) Random Sampling (b) Network Expansion

Figure 3: Two naïve location anonymization models.

2.4 Two Motivating Anonymization Models

The focus of this work is to develop a robust and scalable location anonymization model. While multiple models are available to perform location anonymization to meet users' privacy requirements, we are interested in the optimal one that leads to low query-processing cost and high inference-attack resilience. Below we present two basic models, based on *random sampling* and *network expansion*, respectively. They achieve either extreme of the spectrum, and motivate our star-graph-based model.

Random Sampling Given a query $\mathtt{q}$ issued by a user $\mathtt{u}$ with service profile $(\delta_k^{\mathtt{u}}, \delta_l^{\mathtt{u}}, \sigma_s^{\mathtt{u}}, \sigma_t^{\mathtt{u}})$, at each iteration, this model samples one segment at random from the spatial region as defined by $\sigma_s^{\mathtt{u}}$ and adds it to $\mathtt{u}$'s anonymous location. The process continues until both requirements $(\delta_k^{\mathtt{u}}, \delta_l^{\mathtt{u}})$ are satisfied. As an example, consider a user $\mathtt{u}$ in Figure 3a with $(\delta_k^{\mathtt{u}}, \delta_l^{\mathtt{u}}) = (5, 5)$. With the Random Sampling model, four segments are randomly selected (containing four active users $\{\mathtt{u}_i\}_{i=1}^4$), in addition to the original one that $\mathtt{q}$ is associated with, as highlighted with bold lines.

Network Expansion At the other extreme of the spectrum, one can perturb $\mathtt{u}$'s location based on the Network Expansion model [26]: starting from the original segment which $\mathtt{u}$ is on, following Dijkstra's algorithm, one incrementally adds in neighboring segments, ordered by their network distances (mid points) to $\mathtt{u}$'s position. The process halts when $\mathtt{u}$'s privacy requirements are met. For example, as shown in Figure 3b, the four segments with the minimum network distances to $\mathtt{u}$'s position are added incrementally to form $\mathtt{u}$'s anonymous location.

Discussion It is observed that under the same requirements (δ_k, δ_l, σ_s), the Random Sampling model results in a set of segments evenly distributed over the spatial region as defined by σ_s, which is essentially equivalent to issuing a set of queries at different locations, thus incurring expensive query-processing cost. The strength of this scheme, however, lies in its robust resilience against inference attacks since the set of segments are selected at random.

In contrast, the Network Expansion model generates a set of segments lying in a tightly compact structure. As will be proved in Section 4, such structure features the minimal query-processing cost (the number of boundary nodes grows sub-linearly with the number of segments); the cost is further reduced by that the expanded network is a partial result in the query processing [26]. Nevertheless, this model suffers low attack resilience as the expansion process follows a best-first search paradigm, which can be potentially exploited by an adversary to perform reverse-engineering attacks.

2.5 XSTAR In Nutshell

Motivated by the strengths and weaknesses of the above two models, we develop XStar, a star-graph-based location anonymization model, aiming at achieving an optimal balance between low query-processing cost and high inference-attack resilience. Specifically, XStar achieves this balance in two main phases. First, it groups neighboring queries into the *cloaking-star* structures. The goal is to carefully choose the set of stars that minimize the computation and communication costs. Second, it adjusts the resulted cloaking-stars, and merges neighboring ones into the *super-star* structures if necessary, to fulfill the privacy requirements of each individual user.

Informally, in XStar, the low query-processing cost (details in Section 4) is guaranteed by the cost-aware star selection scheme, and the compact structure of the anonymous location; meanwhile, the high attack resilience (details in Section 5) is contributed by the cloaking-star based perturbation, and the randomness in the operation of merging neighboring stars.

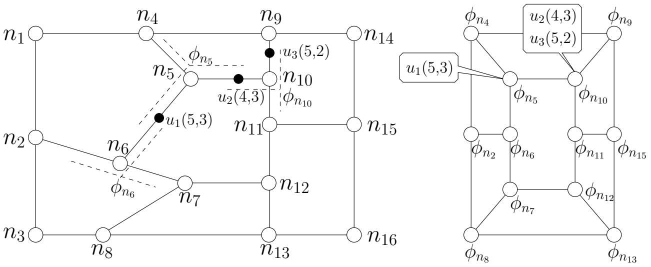


Figure 4: Illustration of XStar model.

Further, in implementing XStar (details in Section 6), we propose a suite of optimization strategies to improve its throughput; also, we introduce *sharing processing of multiple queries* (MQS) for the service provider, which brings considerable performance enhancement.

3. ANATOMY

The location anonymization operation of XStar comprises two main phases, cloaking-star selection and super-star construction. Briefly, in the first phase, a set of neighboring queries are grouped into a cloaking-star structure returned by a cost-aware star selection scheme; in the second phase, the privacy requirements of individual users are imposed by merging a set of neighboring stars into a super-star structure. The details of the two phases are presented in Section 3.1 and Section 3.2, respectively.

3.1 Cloaking-Star Selection

We first introduce the concept of *cloaking-star* that serves as the fundamental structure for location anonymization in XStar.

Definition 5 (Cloaking-Star). *For an intersection node n in the network G, the cloaking-star ϕ_n is the subgraph of G comprised of n and all segments adjacent to n.*

By this definition, every node n with $\mathsf{d}_G(n) \geq 3$ is associated with a unique star ϕ_n. For example, in the left plot of Figure 4, the star ϕ_{n_5} consists of the node n_5 and the segments $\{\overline{n_5 n_4}, \overline{n_5 n_6}, \overline{n_5 n_{10}}\}$.

The cloaking-star structure possesses several desired properties for our purpose. (1) It preserves the locality of neighboring segments; therefore, employing it as the basic unit of anonymization is expected to lead to anonymous locations with highly compact structures. (2) It is amenable to indexing, for the node identifier suffices to represent a star without information loss; using it to represent the anonymous location can alleviate the communication cost, and simplify the implementation.

Given a road network $G = (\mathcal{V}_G, \mathcal{E}_G)$, one can construct a corresponding *star network* $G_\phi = (\mathcal{V}_{G_\phi}, \mathcal{E}_{G_\phi})$, wherein each node represents a star in G, and two nodes are adjacent if their corresponding stars in G share a common segment. The right plot of Figure 4 shows the star network corresponding to the left road network. Note that in G_ϕ, all the edges are of unit length. The distance between two stars ϕ_i and ϕ_j in a road network G is defined as their network distance in G_ϕ, called *hop*, denoted by $\mathsf{h}_G(\phi_i, \phi_j)$. For example, in Figure 4, $\mathsf{h}_G(\phi_{n_6}, \phi_{n_{10}}) = 2$, since their shortest path in G_ϕ is composed of ϕ_{n_6}, ϕ_{n_5}, and $\phi_{n_{10}}$.

A segment is marked as *active* if it is associated with at least one active query. To make our model resilient against the inference attack (Section 5) and amenable to the sharing processing of multiple queries (Section 6), all the queries on the same segment share the same anonymous location.

If a star ϕ is chosen as the anonymous location for the queries on certain active segment s, it is said that ϕ is "selected", and s is "assigned" to ϕ, denoted by $s \leftarrow \phi$. Consider a segment s with two ends as n_0^s and n_1^s. If both $\mathsf{d}_G(n_s^s) \geq 3$ and $\mathsf{d}_G(n_t^s) \geq 3$ hold, then s is associated with two stars $\phi_{n_s^s}$ and $\phi_{n_t^s}$, e.g., ϕ_{n_5} and ϕ_{n_6} for $\overline{n_5 n_6}$ in Figure 4. In this case, it is to be determined to which star s should be assigned, $\phi_{n_0^s}$ or $\phi_{n_1^s}$. In a sequel, over the whole network, one needs to select a set of stars Φ to cover all the active segments.

To achieve low query-processing cost, it is desired to incorporate the cost model into this selection process. Formally, let $\mathsf{cost}(\phi)$ be the cost of executing a typical query with the anonymous location as ϕ (as will be discussed in Section 4), $\mathcal{AS}$ denote the set of current active segments in the road network, and Φ be the set of selected stars, then the problem of minimizing the overall cost can be formalized as

$$\begin{array}{ll} \min\limits_{\Phi} & \sum_{\phi \in \Phi} \mathsf{cost}(\phi) \\ \text{s.t.} & \forall s \in \mathcal{AS}, \exists \phi \in \Phi, s \leftarrow \phi \end{array}$$

This cost-aware segment-to-star assignment scheme aims at finding a set of stars Φ that cover all the active segments, with the lowest overall cost. We note that this modeling ignores the numbers of queries on active segments for simplicity; as indicated in the empirical evaluation (Section 7), however, it has captured the essential elements of the overall cost of query processing at the server, especially after introducing the machinery of *sharing processing of multiple queries* (Section 6).

Unfortunately, no efficient solution exists to this optimization problem, unless P = NP, as shown in the next theorem (the proof is referred to our technical report [28]).

THEOREM 2. *Reductible from the* **weighted vertex cover** *problem, this cloaking-star selection problem is NP-Hard.*

Consequently, instead of attempting to find a global optimal solution, we propose an efficient randomized algorithm that can find high-quality approximate solutions, and is robust against inference attacks.

Specifically, the procedure of inserting a new arrival query q associated with a segment s ($\mathsf{InsertQuery}$) is outlined in the following four cases: (1) if certain star already covers s, the algorithm halts; (2) if both stars $\phi_{n_0^s}$ and $\phi_{n_1^s}$ are already selected, yet s is not covered, one assigns s to one of the two stars with probability in reverse proportion to their corresponding costs; (3) if only one of $\phi_{n_s^s}$ or $\phi_{n_t^s}$ has been selected, s is assigned to that star; (4) if neither $\phi_{n_s^s}$ nor $\phi_{n_t^s}$ is selected, one assigns s to one of them with probability reversely proportional to the corresponding cost.

Essentially, this algorithm ensures that an active segment s is assigned to $\phi_{n_0^s}$ with probability $\mathsf{cost}(\phi_{n_1^s})/[\mathsf{cost}(\phi_{n_0^s}) + \mathsf{cost}(\phi_{n_1^s})]$, or $\phi_{n_1^s}$ otherwise. This property guarantees that the quality of the selected star set Φ does not deviate far from the optimal one, as shown in the next theorem (the proof is referred to our technical report [28]):

THEOREM 3. *Let cost^{opt} be the cost achieved by the optimal star set. The randomized star-selection algorithm achieves the cost cost^{rnd} satisfying $\mathbb{E}\left[\mathsf{cost}^{rnd}\right] \leq 2 \cdot \mathsf{cost}^{opt}$.*

It is worth emphasizing that the quality of the selected stars does not degrade with continuous insertion and deletion of queries, given the fact that it makes no assumption regarding the arrival order of the queries, which is a desired feature in supporting real-time road-network-based LBS.

3.2 Super-Star Construction

In the previous phase, a set of stars are selected to cover active segments, with the criterion of query-processing cost. In this phase, we fulfill the privacy requirements of mobile users. This objective is achieved by merging a set of neighboring stars to form a *super-star* structure, which then serves as the anonymous location for the queries inside.

DEFINITION 6 (SUPER-STAR). *A set of stars $\{\phi_i\}_{i=1}^{|\psi|}$ is said to form a super-star ψ if the subgraph comprising $\{\phi_i\}_{i=1}^{|\psi|}$ is connected, where $|\psi|$ denotes the number of stars in ψ (the cardinality of ψ).*

As an example, in Figure 4, the user u_1 is assigned to the star ϕ_{n_5}, while u_2 and u_3 are assigned to $\phi_{n_{10}}$. However, the numbers of segments and active users in ϕ_{n_5} or $\phi_{n_{10}}$ alone do not satisfy users' privacy requirements (δ_l^u, δ_k^u) as $(5, 3)$, $(4, 3)$, and $(5, 2)$, respectively. By merging ϕ_{n_5} and $\phi_{n_{10}}$, one obtains a super-star ψ that meets the requirements of all users involved.

Recall that the service profile of a user u is specified as a quadruplet $(\delta_k^{\mathsf{u}}, \delta_l^{\mathsf{u}}, \sigma_s^{\mathsf{u}}, \sigma_t^{\mathsf{u}})$, corresponding to k-anonymity, l-diversity, spatial tolerance, and temporal tolerance. In out setting, particularly, in a super-star ψ, σ_s^{u} is defined in terms of the hop count between the star covering u and the furthest star in ψ; and if a query can not be successfully anonymized within σ_t^{u}, it is discarded.

Now, we sketch the procedure of merging stars to form a super-star structure ($\mathsf{MergeStar}$): starting from an initial star ϕ, one applies a bottom-up aggregation, incrementally adding in neighboring stars until the privacy requirements of all users inside the super-star are satisfied, or the spatial tolerance of certain user is reached. More concretely, one first checks if the star ϕ already satisfies the privacy requirements of users in it; if not, one iteratively adds in neighboring active stars if possible. At each iteration, one first identifies all the neighboring stars whose fusion with current super-star ψ do not violate the spatial tolerance of any user inside; if such a star exists, one randomly picks one to merge with ψ to form a new super-star. This expansion process iterates until meeting the privacy requirements of all the users inside ψ, or reports failure (all the involved queries will be pushed back to the queue, awaiting for anonymization triggered by new arrival queries).

4. QUERY PROCESSING COST

The cost of processing a location-based query consists of both the query execution cost at the LBS provider, and the communication cost of transferring the query result back to the mobile client on the move. In this section, we establish an analytical model to compare the three location anonymization models proposed in this paper from the perspective of query-processing cost.

4.1 Cost Measures

Query Execution Cost Let $\mathcal{C}_s$ and $\mathcal{C}_n$ denote the computation cost (in terms of both CPU and IO) of a segment-based operation and a node-based one, respectively. Note that both $\mathcal{C}_s$ and $\mathcal{C}_n$ may vary from segment to segment or from node to node, depending on the condition of the road network (e.g., the density of objects), the predicate of each query (e.g., the parameter k of a k-NN query), and the system implementation (e.g., the performance of the look-up table). In the prototype system of XStar, we set $\mathcal{C}_s$ and $\mathcal{C}_n$ statically for a typical setting (a detail discussion in Section 7). One direction of our ongoing research is to develop finer granularity and dynamic cost models.

Hence, for a typical query with a set of segments S as its anonymous location, the execution cost, $\mathsf{cost}_{exec}(S)$, can be approximately estimated as follows: $\mathsf{cost}_{exec}(S) = \mathcal{C}_n \cdot |\mathcal{BV}_S|$ $+ \mathcal{C}_s \cdot |S|$, where $|\cdot|$ denotes the cardinality of the set.

Communication Cost Next, we analyze the additional communication cost incurred by the location anonymization operation. We measure the communication cost in terms of the length of the sent and received messages. Recall the framework of privacy-aware mobile service in Section 2. For given privacy requirements, i.e., a stable number of segments in the anonymous location, the cost of sending and receiving the candidate answer becomes the dominant communication cost between the location anonymization engine and the LBS provider.

Given a k-NN query q and a set of segments S as the anonymous location, $|\mathcal{R}(\mathsf{q}, S)|$ is estimated as: $|\mathcal{R}(\mathsf{q}, S)| \approx k \cdot |\mathcal{BV}_S| + \sum_{s \in S} |\mathcal{O}(\mathsf{q}, s)|$, where the first component corresponds to the result size of issuing a k-NN query over each boundary node of S, and the second represents all the objects on the segments of S.

Let ρ_o denote the average number of objects on a segment, and $\mathcal{C}_o$ be the cost of sending and receiving an object. The communication cost for an anonymous k-NN query, with S as the anonymous location, is estimated as: $\mathsf{cost}_{comm}(S) = \mathcal{C}_o \cdot [k \cdot |\mathcal{BV}_S| + \rho_o \cdot |S|]$.

4.2 Cost Analysis

We now analyze the impacts of the three anonymization models over the query processing overhead. For ease of exposition, we consider a uniform grid as the underlying road network, and assume that all the static objects of interest and active mobile users are distributed over the network with average density of ρ_o and ρ_u per segment. Consider a typical query q with (δ_k, δ_l) specified as location anonymity and diversity. Its anonymous location thus comprises a set of segments S, with $|S| = \max(\delta_k/\rho_u, \delta_l)$. It is noticed that for fixed $|S|$, the size of the boundary node set $|\mathcal{BV}_S|$ becomes the dominant factor in the two cost models above. Therefore, in the following discussion, we focus on analyzing $|\mathcal{BV}_S|$ for each model.

Random Sampling With the assumptions above, the Random Sampling model results in a set of segments S with a boundary node set of size $2|S|$ in the worst case, where no two selected segments are adjacent. Clearly, both the query execution and communication costs grow linearly with the number of segments $|S|$.

Network Expansion In contrast, the Network Expansion model generates a set of segments S with $|\mathcal{BV}_S| = \sqrt{|S|+2}$ in the worst case. Given the fact that the cost of node-based operation usually dominates the computation, i.e., $\mathcal{C}_n \gg \mathcal{C}_s$, here the execution cost grows sub-linearly, square-root-wise, more precisely, with the cardinality of S.

XStar We have shown in Section 3.1 that the cost-aware star selection scheme guarantees the low overall cost of the set of cloaking-stars. Here, we concentrate our analysis on the second phase of XStar, super-star construction.

To be attack resilient, the $\mathsf{MergeStar}$ operation picks neighboring active stars at random, without considering any cost metrics. However, the resulted anonymous location usually exhibits desired properties in terms of query processing cost: (1) the star structure preserves the locality of neighboring segments, i.e., in a star, the number of boundary nodes is no more than that of involved segments; (2) the star selected at each iteration must satisfy the user-specified maximum spatial tolerance requirement, thereby leading to a fairly compact super-star structure.

It can be proved that the XStar model produces a super-star ψ (with segments as $|S|$) with $|\mathcal{BV}_S|$ no more than $(2|S|+4)/3$, i.e., approximately one third of that by the Random Sampling model. Also note that this worst case occurs only at the extreme condition when the stars forming ψ lay in a chain, with probability lower than $(1/4)^{|S|/3}$. In real applications, as verified by our experiments, XStar usually generates anonymous locations with quality comparable to that achieved by the Network Expansion model.

5. INFERENCE ATTACK RESILIENCE

The obfuscation of the exact location information is only a part of the story, one needs to consider the resilience of the anonymization against the adversary's attack: based on her prior knowledge or understanding regarding the working anonymization model, the adversary attempts to reveal users' original location through the blurred information. We note that the attack discussed here focuses on one-shot queries. We believe that by exploiting location information associated with multiple queries issued by a single user, the adversary can significantly improve her chance of pinpointing the user. Addressing privacy breach in (continuous) multiple queries is one direction of our ongoing research.

Given the anonymous location as a set of segments S, the ideal protection is achieved if each segment is indistinguishable to the adversary; from her perspective, the mobile user is associated with each segment in S with equal probability $1/|S|$. However, with effective attacks, the adversary can identify that the association between u and a specific segment $s \in S$ has much higher probability than $1/|S|$, thereby revealing u's private location with high confidence. We capture such vulnerability using the notion of *Linkability*:

Definition 7 (Linkability). *Given a user* $\mathtt{u}$ *with exact location as a segment* s^* *and anonymous location as a set of segments* S*. The linkability* $\mathsf{link}[\mathtt{u} \leftarrow s^* | S, K_{ad}]$ *is the probability that an adversary can infer* $\mathtt{u}$*'s association with* s^**, based on* S *and her background knowledge* K_{ad}*.*

In particular, the background knowledge K_{ad} considered in this work includes (1) the location anonymization model, (2) the underlying road network, and (3) the estimation of query-processing cost for every cloaking-star (for XStar only). Following, we present a general *replay attack* model, which serves to measure the attack resilience of the three location anonymization models.

5.1 Replay Attack

In the replay attack, for each segment $s \in S$, by re-running the anonymization algorithm with s assumed to be the original location, the adversary estimates the likelihood of s to generate the anonymous location S, $\mathsf{like}[S|\mathtt{u} \leftarrow s, K_{ad}]$. Under this model, the linkability is calculated as

$$\mathsf{link}[u \leftarrow s^*|S, K_{ad}] = \frac{\mathsf{like}[S|\mathtt{u} \leftarrow s^*, K_{ad}]}{\sum_{s\in S}\mathsf{like}[S|\mathtt{u} \leftarrow s, K_{ad}]}$$

Specifically, we assume that the adversary has full knowledge regarding the anonymization algorithm, $\mathcal{A}(\cdot)$, which takes a segment s as input, and generates a set of segments as the anonymous location for s. Therefore, she is able to *replay* the anonymization process: for each $s \in S$, (1) run $\mathcal{A}(s)$, and generate a set of segments S', with $|S'| = |S|$; (2) compute the likelihood $\mathsf{like}[S, |\mathtt{u} \leftarrow s, K_{ad}] = |S' \cap S|/|S|$; and (3) select the segment s^+ that leads to the largest likelihood value as the original location, $s^+ = \arg\max_s \mathsf{like}[S|\mathtt{u} \leftarrow s, K_{ad}]$.

5.2 Resilience Analysis

Next, we proceed to analyzing the resilience of the three location anonymization models with respect to the replay attack.

Random Sampling Under this model, the extra $(|S|-1)$ segments are selected at random; therefore, following the replay attack model, the adversary will find that each segment $s \in S$ can generate S with identical probability, i.e., $\max_s \mathsf{link}[\mathtt{u} \leftarrow s|S, K_{ad}] = 1/|S|$, which implies the possible strongest protection.

Network Expansion Under this model, the $(|S|-1)$ extra segments are expanded from the original one s^*, based on their network distances to s^*. With the replay attack, the adversary runs the network expansion algorithm for each $s \in S$. Clearly, s^* will generate $S' = S$, i.e., the highest likelihood, while other segments tend to result in likelihood less than that by s^*, therefore highlighting s^* as the expansion source. This is empirically verified in our experiments.

XStar Recall that given the original segment s^*, the first phase of XStar generates a star ϕ covering s^*, and the second phase expands from ϕ and produces a super-star ψ.

Assume that ϕ consists of the segments $\{s_i\}_{i=1}^n$, with the corresponding neighboring stars as $\{\phi_i\}_{i=1}^n$. According to the design principle of the star selection scheme, the likelihood that $\mathtt{u}$ is associated with s_i, given ϕ as $\mathtt{u}$'s anonymous location, $\mathsf{like}[\phi, |\mathtt{u} \leftarrow s_i, K_{ad}]$, is calculated as

$$\mathsf{like}[\phi|\mathtt{u} \leftarrow s_i, K_{ad}] = \frac{\mathsf{cost}(\phi_i)}{\mathsf{cost}(\phi) + \mathsf{cost}(\phi_i)}.$$

Furthermore, the posterior probability $\mathsf{prob}[\mathtt{u} \leftarrow s_i|\phi, K_{ad}]$ is given by:

$$\mathsf{prob}[\mathtt{u} \leftarrow s_i|\phi, K_{ad}] = \frac{\mathsf{like}[\phi|\mathtt{u} \leftarrow s_i, K_{ad}]}{\sum_{j=1}^n \mathsf{like}[\phi|\mathtt{u} \leftarrow s_j, K_{ad}]}$$

It is observed that the adversary can identify the association between $\mathtt{u}$ and a specific segment s_i with high probability only if the cost of ϕ_i and other stars are extremely biased, i.e., $\mathsf{cost}(\phi_i) \gg \mathsf{cost}(\phi_j)$ for all $j \neq i$, which however is unusual in real scenarios, as verified in our experiments.

Now, consider the second phase, provided the facts that (1) ψ is generated by randomly expanding from some initial star, and (2) all the stars in ψ contain active users, and each can initiate and lead to the construction of ψ. Without further knowledge, from the perspective of the adversary, $\mathtt{u}$ is associated with each star with identical probability. Formally, assume that ψ consists of the stars $\{\phi_i\}_{i=1}^m$. The probability that the adversary can infer that $\mathtt{u}$ belongs to ϕ_i given ψ follows: $\mathsf{prob}[\psi|\mathtt{u} \leftarrow \phi_i, K_{ad}] = 1/m$.

Combining the results above, we can now estimate the linkability under the XStar model: $\mathsf{link}[\mathtt{u} \leftarrow s^*|\psi, K_{ad}] = \sum_{\phi\in\psi} \mathsf{prob}[\mathtt{u} \leftarrow s^*|\phi, K_{ad}] \cdot \mathsf{prob}[\mathtt{u} \leftarrow \phi|\psi, K_{ad}]$, where if $s^* \notin \phi$, $\mathsf{prob}[\mathtt{u} \leftarrow s^*|\phi, K_{ad}] = 0$.

The analysis above is empirically verified in Section 7. It is also shown that in real road networks, the XStar model provides almost the same amount of resilience as the Random Sampling model against the replay attack.

6. IMPLEMENTATION

In this section, we deal with the detailed issues of implementing XStar in the location anonymization engine (LAE), and propose multiple optimizations to improve its performance. Further, we boost the throughput of the query processing server by introducing the strategy of multiple queries sharing processing (MQS).

6.1 Location Anonymization Engine

Algorithm 1: Location Anonymization Engine

```
   // $Q_q$:arrival-query queue, $H_q$:expiration heap
   // $I_\phi$:active-star index, $Q_\phi$:ready-star queue
1  while $true$ do
2      $Q_\phi \leftarrow \emptyset$;
       // purge of expired requests
3      while $true$ do
4          q ← top entry of $H_q$;
5          if $q$ not expired then break;
6          $\phi \leftarrow$ DeleteQuery(q);
           // $\phi$ still active
7          if $\phi \in I_\phi$ then add $\phi$ to $Q_\phi$;
       // insertion of new requests
8      if $Q_q \neq \emptyset$ then
9          q ← first entry of $Q_q$;
10         $\phi \leftarrow$ InsertQuery(q);
11         add $\phi$ to $Q_\phi$;
       // location anonymization
12     while $Q_\phi \neq \emptyset$ do
13         $\phi \leftarrow$ first entry of $Q_\phi$;
14         MergeStar($\phi$);
```

Algorithm 1 sketches the main procedure of LAE: at each iteration, it first purges all expired requests, and pushes the affected active stars to the ready-star queue Q_ϕ to be processed (line 3-7); it then pops up one new request q from the query queue Q_q, and pushes the selected star ϕ to Q_ϕ (line 8-11); finally, it attempts to perform anonymization for each star in Q_ϕ (line 12-14).

6.2 Optimizations

Despite its simplicity, the basic version of LAE introduced above suffers several drawbacks. (1) Each request-deletion operation (DeleteQuery) results in a trial of anonymizing the affected star, without checking the success condition. (2) It attempts to anonymize the affected star immediately after a new query is inserted (InsertQuery). It is expected that a significant number of attempts would fail because of insufficient numbers of active users or segments to satisfy users' privacy requirements. (3) For each request, the anonymization process starts from the scratch in a bottom-up manner, thereby incurring the scalability problem.

In the following, corresponding to the above drawbacks, we present multi-folded optimizations to improve the success rate and scalability of LAE.

Lazy Update for Deletion We propose this policy based on the following observation: if the anonymization of a star ϕ failed in a previous iteration, and no updates happen to other stars, then ϕ can possibly be anonymized only if updates occur to its profile parameters $(\delta_d^\phi, \delta_k^\phi, \sigma_s^\phi)$, which are defined as the maximum k-anonymity, maximum l-diversity, and minimum spatial tolerance values associated with the queries in ϕ, respectively. Therefore, on deleting a query (DeleteQuery), one attempts to anonymize the affected star ϕ only if its profile $(\delta_d^\phi, \delta_k^\phi, \sigma_s^\phi)$ is updated.

Batch Insertion of Queries To improve the success rate of the anonymization operation, at each iteration, one can insert a batch of new queries (InsertQuery), i.e., waiting for a period of time T_w, before beginning the anonymization process. The parameter T_w can be adjusted to trade the average processing burden over LAE for the success rate of the anonymization operation. Also, T_w should be set according to users' maximum tolerable service delay. The optimal setting of T_w is discussed in Section 7.

Early Failure Detection The merging-star (MergeStar) operation is costly in that it might fail because of insufficient segments or active users appearing in the neighborhood of the initial star ϕ. It is thus desired to maintain such statistical information, and stop the merging process early if detecting no enough number of active users or segments.

Specifically, for each intersection node n, we maintain the number of active users $\mathsf{num}_u(n, r)$ and segments $\mathsf{num}_s(n, r)$ in the sub-network within radius r hops to n. The statistical information for multiple r's $\{r_i\}_{i=0}^h$ (with r_0 corresponding to the n-centered star) can be cached in order to achieve more effective detection, though at higher maintenance cost.

The cached information can be used in two ways. (1) On anonymizing an initial star ϕ, with center node n, and profile $(\delta_l^\phi, \delta_k^\phi, \sigma_s^\phi)$, check if $\exists i \in [0, h]$ such that (i) $r_i \geq \sigma_s^\phi$, and (ii) $\mathsf{num}_u(n, R_i) < \delta_k^\phi$ or $\mathsf{num}_s(n, L_i) < \delta_l^\phi$. If such i exists, then the merging process stops as failure. (2) On adding a new n'-centered star ϕ' to the current super-star ψ, check if $\exists i \in [0, h]$, such that (i) $r_i \geq \sigma_s^{\phi'}$, and (ii) $\mathsf{num}_u(n', R_i) < \max\{\delta_k^{\phi'}, \delta_k^\psi\}$ or $\mathsf{num}_s(n', R_i) < \max\{\delta_l^{\phi'}, \delta_l^\psi\}$. If such i exists, one can safely exclude ϕ' from the candidate list for expansion.

Note that the performance enhancement is not achieved at the cost of compromising privacy guarantees; the optimization strategies introduced above adhere to the hard privacy requirements specified by users; without further information, e.g., the arrival order of queries at LAE, the performed optimizations offer no leverage to an adversary.

6.3 Sharing Processing Of Multiple Queries

From the perspective of anonymous query processing, XStar enjoys two major advantages over conventional models. (1) Independence of the underlying query processing techniques. XStar is optimizable for any specific implementation, as discussed in Section 4, by configuring the cost function according to the adopted model, e.g., solution indexing [20] or network expansion [26]. (2) Capability of sharing processing of multiple queries. It considers the possibility of sharing processing in the location privatization operation, by grouping queries with nearby locations together and perturbing their location as an entirety.

Concretely, given a set of k-NN style queries $\{\mathsf{q}_i\}_{i=1}^t$ sharing the same anonymous location ψ, with corresponding k's as $\{k_i\}_{i=1}^t$ and $k_i \leq k_j$ for $i < j$, one can evaluate q_t once and retrieve the top k_i objects of $\mathcal{O}(\mathsf{q}_t, n)$ as $\mathcal{O}(\mathsf{q}_i, n)$ for each $n \in \mathcal{BV}_\psi$ and $i \in [1, t-1]$. This principle can also be extended to other query types, e.g., range queries; the details are omitted due to the space limitation.

7. EVALUATION

In this section, we perform an empirical analysis of the location anonymization models proposed in the paper. The experiments are designed to compare these models based on the following three metrics. (1) Cost awareness. The protection mechanism should not incur excessive system burden for either the service provider or the mobile client, in terms of query processing and communication costs. (2) Attack resilience. The applied protection should be robust against malicious inference attacks; that is, it is difficult for the adversary to penetrate the protection to identify users' exact positions with high certainty. (3) Operation efficiency. The anonymization operation should be computationally efficient and scalable; a location anonymization engine equipped with modest computational resources should be able to handle a large number of mobile users on continuous move.

7.1 Experimental Setting

All our experiments were performed over real road maps from areas of the United States. The first road map corresponds to the highways of the entire State of California [23], which contains 21,048 nodes and 21,693 edges; moreover, it is associated with a real dataset of 104,771 *points of interest*, as categorized into 62 classes, e.g., church, hospital, airport, etc., which we used as queried objects in our simulation. The second road map corresponds to the roads in the City of Oldenburg, which contains 6,105 nodes and 7,035 edges. Choosing these two road maps, we intend to evaluate the performance of location anonymization models for road networks at varying scales.

On these maps we simulated different traffic conditions using the Network-based Generator of Moving Objects by T. Brinkhoff[2], a state-of-the-art traffic simulator. We assign a same number (10,000) of moving objects to each map. Since the two maps are of significantly different scales, we intend to simulate both high user density (rush hour) and low user density (non-rush hour) conditions. In each simulation, we defined two classes of moving objects, with speeds corresponding to slow (e.g., trucks) and fast (e.g., passenger cars) vehicles, respectively. With a randomly assigned probability, each vehicle generates a set of (or none) k-NN queries during the simulation, with the parameters specified as follows: (1) the requested number of nearest points of interest as k; (2) the category of the points of interest as c, e.g., church, hospital, etc.; (3) the privacy requirements as k-anonymity (δ_k) and l-diversity (δ_l); and (4) the service quality requirements as the spatial (σ_s) and temporal tolerance (σ_t). The values of each query are drawn inde-

[2] http://www.fh-oow.de/institute/iapg/personen/brinkhoff

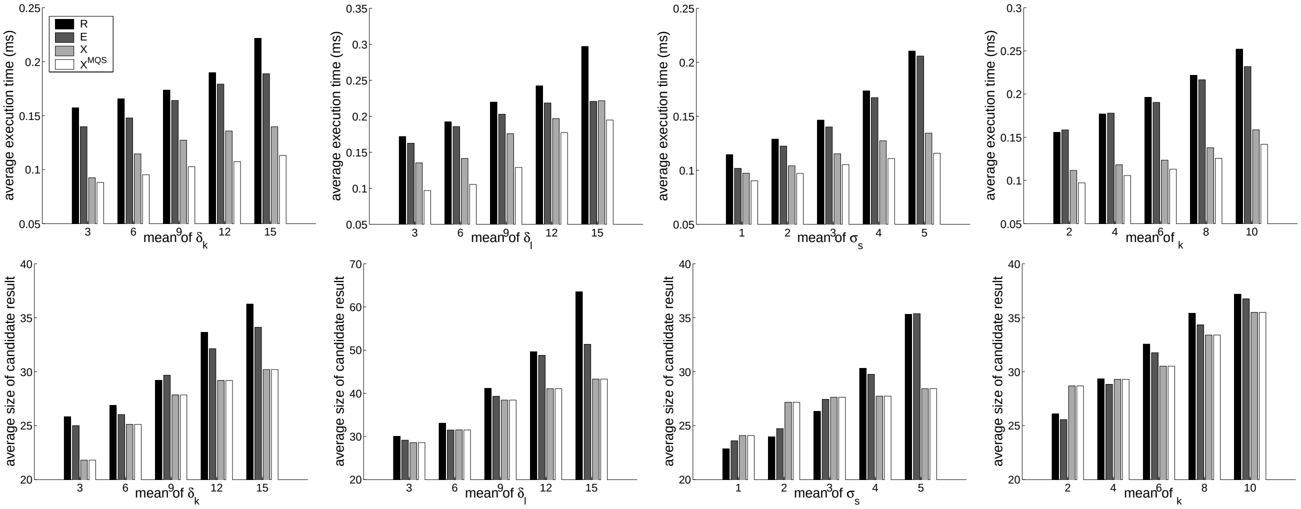


Figure 5: Query execution cost (in terms of the average execution time per query) and communication cost (in terms of the average size of candidate result per query) with respect to varying settings of parameters δ_k, δ_l, σ_s, and k.

pendently following certain distributions, with parameters listed in Table 1. After issuing a query, the vehicle waits for some normally distributed inter-wait time γ, until the request is either answered or dropped, before issuing another service request.

parameters	k	c	δ_k	δ_l	σ_s	σ_t	γ
mean	5	N/A	5	5	4	10	20
deviation	1	N/A	1.5	1.5	1	2	2

Table 1: Default parameter setting for query generation. Note: all the parameters except c follow normal distributions; c follows a uniform distribution over the interval [0, 62]; the values of σ_t and γ are in the unit of second.

For the location anonymization engine, we implemented four different methods: Random Sampling (R), Network Expansion (E), basic XStar (X), and optimized XStar (X*). For the anonymous query processing server, two versions were developed, one with the machinery of multiple queries sharing processing (MQS), and the other without MQS. Both the location anonymization engine and the query processing server were implemented in Java. The experiments were performed on a Linux box running 1.5Ghz CPU with 512 MB memory.

7.2 Experimental Results

Cost Awareness

In the first set of experiments, we take a close examination of the impact of the location anonymization operation over the service performance, i.e., the query execution cost and the communication cost, under varying setting of traffic condition, and privacy (δ_k, δ_l) and service (σ_s, k) requirements. Specifically, in terms of the query execution cost, we measure the average execution time of processing a query at the server side; while in terms of the communication cost, we measure the average number of objects returned in the candidate result of a query; in the experiments, we use the road map of California and its associated dataset of points of interest in query processing.

We measure the query processing cost corresponding to the three location anonymization models (R, E, and X represent the Random Sampling, Network Expansion and XStar methods, respectively) in the case without multiple queries sharing processing (MQS), and the XStar model with MQS policy ($\mathsf{X}^{\mathsf{MQS}}$), as the mean values of the parameters k, δ_l, δ_k, and σ_s vary within the intervals of [2, 10], [3, 15], [3, 15] and [1, 5], respectively. In each set of experiments, we fix three parameters and vary the last one. The upper row of Figure 5 plots the average execution time of processing an anonymous query by the server, while the bottom row plots the average size of the candidate result per query returned by the server.

With respect to the anonymous query execution time, it is observed that R-scheme incurs the highest system overhead at the server among the three schemes and X-scheme outperforms both R and E in most cases, which validates our theoretical analysis regarding its superiority in terms of the query execution cost. Note that although working ideally in the homogeneous grid world (Section 4), E-scheme ignores the heterogeneity of real road maps when selecting extra segments, resulting in its unsatisfying performance in real applications. Also, notice that the MQS policy improves the query processing efficiency, and the improvement tends to become significant as the parameters δ_l, δ_k, k, or σ_s increases. This can be explained by the following facts. (1) Stricter privacy requirements δ_l and δ_k cause a larger number of queries to be anonymized in batches; therefore, more queries can be processed in groups. (2) A larger k results in higher execution cost for each individual query, but also offers more considerable savings for grouped queries. (3) A large spatial tolerance σ_s boosts the chance for a query to be successfully anonymized (to satisfy δ_k and δ_l) by allowing more queries to be grouped together.

Moreover, by examining the impacts of these four parameters over the query execution cost, one can notice that the parameters δ_k and δ_l have stronger influence than σ_s and k for all the anonymization models. This is contributed by the fact that stricter privacy requirements lead to anonymous locations of much coarser granularity (larger area), with its impact over the query execution easily exceeding

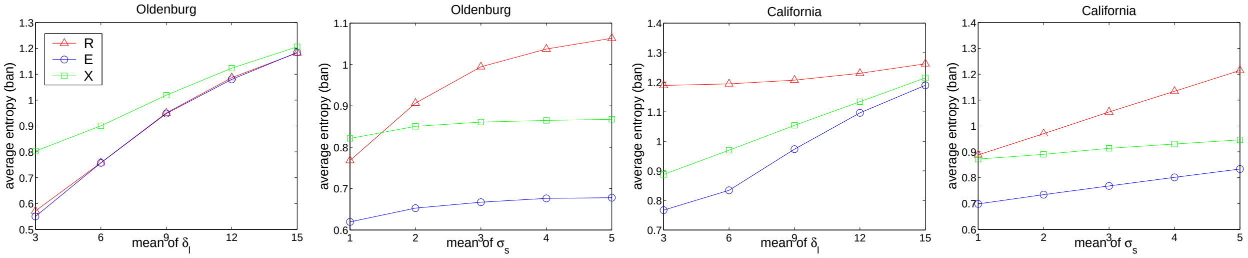


Figure 6: Average information entropy of anonymous locations generated by location anonymization models with respect to δ_l and σ_s, for maps of Oldenburg and California. Note: the entropy is in unit of ban (Hart).

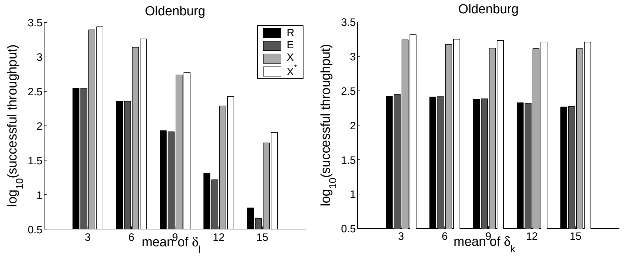


Figure 7: Successful throughput with respect to δ_l and δ_k.

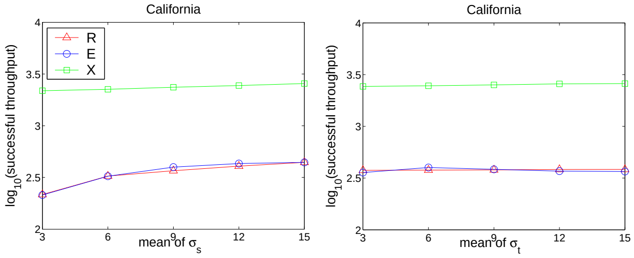


Figure 8: Successful throughput with respect to σ_s and σ_t.

that exerted by the requested number of points of interest k or spatial tolerance σ_s. Meanwhile, it is also interesting to notice that the performance of X-scheme is fairly insensitive to the parameter setting: in all four cases, its increase is the least significant among all the anonymization models under consideration.

With respect to the communication cost, as expected, R-scheme generates significantly larger size of candidate result than the others (note that the multiple queries sharing processing does not affect the average size of candidate result; therefore, X and $\mathsf{X}^{\mathsf{MQS}}$ have the same size). Meanwhile, X-scheme outperforms the other two for the cases of varying δ_k and δ_l, though the lead is not considerable as that in terms of the query execution time, especially for small δ_k and δ_l. In the cases of varying σ_s and k, it is observed that both R and E schemes perform slightly better than X for small σ_s and k. All these phenomena are explained by the following fact: both R and E perform segment-based perturbation, which stop just after obtaining a sufficient number of segments; meanwhile, X performs star-based perturbation, and the generated anonymous location may include slightly more than enough number of segments (or nodes). This difference exhibits significantly when these parameters are small; however, as the privacy or service quality requirements grow higher, the inherent superiority of star-based perturbation dominates the performance.

Attack Resilience

Now, we proceed to evaluating the resilience of the anonymous locations generated by different anonymization models against malicious inference attacks. Specifically, we consider the replay attack as described in Section 5: given a set of segments S as the user u's anonymous location, the adversary attempts to estimate for each segment $s \in S$ its probability of being associated with u. We measure the strength of the privacy protection using the information entropy of the distribution of such probabilistic estimation; a larger entropy value indicates higher uncertainty for the adversary, i.e., better protection. We use both the California and Oldenburg road networks, aiming at capturing the influence of factors such as area scale and user density.

The first set of results is illustrated in Figure 6, where we measure the information entropy of anonymous locations, with respect to varying δ_l and σ_s, two parameters relevant to the spatial expansion of anonymous locations. The left two plots correspond to the road network of Oldenburg, and the right two that of California. First notice that the protection strengths of all the models increase as segment diversity (δ_l) or spatial tolerance (σ_s) grows; intuitively, an anonymous location containing more segments tends to provide better protection. Also, as expected, under the replay attack, the protection provided by E is easy to penetrate, while R demonstrates the best protection strength in most cases. The performance of X is fairly stable, and its difference with R tends to decrease as the number of segments increases. For the case of Oldenburg road network (the leftmost plot), the entropy corresponding to X is even higher than that provided by R under varying settings of δ_l. This can be explained by that for a road network with sufficient user density, and for given privacy and service quality requirements, the anonymous location generated by a star-based perturbation scheme tends to feature higher segment-diversity than that produced by segment-based perturbation schemes, yet without compromising query processing efficiency.

Also, note that the attack resilience discussed here focuses on the case of one-shot query. We anticipate that by combining the location information of multiple continuous queries issued by a particular mobile user, the adversary can potentially infer more positioning information, which we consider as a valuable research direction of our future work.

Operation Efficiency

The last set of experiments are designed to evaluate the operation efficiency of various location anonymization models.

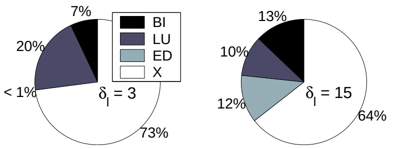


Figure 9: Fractions of improvement contributed by the multiple optimization strategies.

In particular, we are interested in two critical measures: the success rate of location anonymization operation, which indicates the responsiveness of the model, and the average execution time of anonymizing a query, which reflects the scalability of the model. We incorporate the two measures into a single metric, *successful throughput* (SF):

$$\mathsf{SF} = \text{query arrival rate} \times \text{anonymization success rate}$$

That is, for a given number of LBS requests, a higher successful throughput indicates better performance of the location anonymization engine.

In our experiments, specifically, in addition to the R, E, and X-scheme discussed so far, we implement an optimized version of X-schem, X^*, which incorporates the optimizations introduced in Section 6. We measure the successful throughput of these four models as functions of the parameters δ_k, δ_l, σ_s, and σ_t (the parameter k is not relevant to query anonymization).

The result is illustrated in Figure 7 and 8. Figure 7 shows the influence of the privacy parameters δ_k and δ_l over the SF of anonymization. Overall, it is noticed that the performance of all four schemes tend to decrease as the privacy requirement becomes stronger. It is expected because (1) larger δ_l and δ_k are harder to satisfy, leading to higher rate of failure, and (2) moreover, even a successful attempt takes longer execution time. The SF of X (and X^*) is significantly higher than that of R and E, and the gap tends to grow as the privacy requirements become stricter. For example, even basic X-scheme maintains throughput at about 56 for $\delta_k = 15$. This can be attributed to the strategy of star-based perturbation: compared with segment-based perturbation, which involves costly distance computation for node (or segment) pairs, the randomized star selection and merging operations are much less costly. It is also observed that the setting of δ_k has less impact than δ_l over SF. Keep in mind, however, that the Oldenburg road network features high user density; therefore, δ_k is easier to satisfy than δ_l.

Figure 8 demonstrates how the setting of spatial and temporal tolerance affects the SF. Clearly, the SFs of all three anonymization models exhibit increasing trends as the spatial tolerance grows. This is expected, since a larger spatial expansion increases the chance for a query to be successfully anonymized. Meanwhile, interestingly, the SFs of all the models stay fairly stable as the temporal tolerance changes. This might be explained as follows: longer lives of queries increase their chance to be successfully anonymized, i.e., a higher success rate of anonymization, but also increases computational overhead for the anonymization engine, i.e., a large stack of stale queries, which exist as two factors with countering influence over the SF.

It is also interesting to analyze the contributions by various optimization strategies to the SF of X^*-scheme. The space limitation precludes the possibility of a detailed analysis. Here, we take two snapshots of the fractions of improvement contributed by each strategy for $\delta_k = 3$ and 15, respectively. For brevity, we use the following short notations: *lazy update for deletion* (LU), *batch insertion of queries* (BI), and *early detection of failure* (ED). Figure 9 shows the result, from which we can obtain the following observations. (1) The contribution from LU generally decreases as δ_l grows. This is explained by that as the success rate of anonymization decreases, more queries tend to be deleted, resulting in frequent changes to the privacy profiles of the corresponding stars; therefore, LU takes less effect in preventing false update. (2) The portion of BI stays stable as the parameters change. This is because the insertion of more queries necessarily increases the average number of queries per star, thus improving the success rate of the anonymization operation. (3) The fraction contributed by ED tends to increase with δ_l. This is due to that stronger privacy requirements bring higher failure probability for an anonymization operation, and ED can effectively counter its impact over the performance, by avoiding unnecessary attempts.

8. RELATED WORK

Location privacy is gaining increasing interests from both the mobile networking [3, 6, 10, 15, 16] and data management [1, 8, 9, 11, 24, 30] communities. The former has been focusing on anonymous routing [10, 21], MIXes in mobile communication systems [6], source location privacy [15, 16], and Mix Zones [2, 3]. Most existing solutions assume the random-waypoint mobility model [14] and utilize location hiding techniques to disable the adversary to associate a location-based service request, be it routing or query, with a particular network entity.

In the data management community, the research on location privacy has been targeting extending data perturbation techniques developed for protecting data privacy to address location privacy concerns. Examples include spatial cloaking [1, 8, 9, 11, 24, 30], false dummies [18], and landmark objects [12]. Arguably, the most popular protection mechanism to date has been spatial cloaking, wherein the exact position of a mobile user is transformed to a spatial cloaking region. The criterion of transformation has been solely based on location anonymity, and the amount of protection has been measured in terms of the regional size of the anonymous location. Unfortunately, as we have pointed out, under the network-constrained mobility model, most existing cloaking techniques tend to fail, because the area size is no longer an effective and valid measure. The work [22] considers privacy-aware query processing for road networks; however, the adopted privacy protection model is still limited to spatial area cloaking.

Processing spatial queries over road networks has been an emerging research topic recently [5, 13, 20, 25, 26]. Two commonly used search paradigms have been proposed for k-NN style queries, namely incremental network expansion [26], and solution indexing [20]. The former gradually expands the search from the query focal point through the edges and reports the accessed objects during the expansion; while the latter pre-computes and caches intermediate results, and constructs the query answer based on the cached results. Another line of research has been directed to continuous nearest neighbor (CNN) query recently [13, 25], in which both the k-NN objects and the valid scopes of the results along a path are returned.

9. CONCLUSION

This paper presents a systematic study on the problem of protecting location privacy under the network-constrained mobility model. We proposed XStar, a general model for privacy-aware mobile services over road networks. Compared with prior work, XStar highlights itself with three distinct features: it supports road-network-specific, personalized privacy and QoS requirements on a per request basis; it strikes a balance between the attack resilience of the performed protection and the processing cost of the anonymous query; it scales to support a large number of mobile users with varying service requirements, through a star-graph-based privatization model, powered by multi-folded optimizations in implementation. Extensive experiments over real road networks have been conducted to evaluate the efficacy of the XStar model.

Our research will continue along several dimensions. First, we plan to develop finer granularity cost models for location-based query evaluation and communication, taking account of dynamic traffic condition and complex road network semantics. Second, we intend to study other types of inference attacks beyond the replay attack model, to evaluate and enhance the attack resilience of XStar. Finally, we are interested in extending the current framework to support continuous location-based queries, which are subjected to much more sophisticated inference attacks, compared with one-shot queries.

Acknowledgement

This work is partially funded by grants from NSF Cybertrust program, an IBM faculty award, an Intel research grant, and an IBM SUR grant.

10. REFERENCES

[1] B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting anonymous location queries in mobile environments with pirvacygrid. In *WWW*, 2008.

[2] A. Beresford. Location privacy in ubiquitous computing. In *IEEE Pervasive Computing*, 2005.

[3] A. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *IEEE PerSec*, 2004.

[4] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. Multi-hop wireless ad hoc network routing protocols. In *MOBICOM*, 1998.

[5] H. Cho and C. Chung. An efficient and scalable approach to cnn queries in a road network. In *VLDB*, 2005.

[6] H. Federrath, A. Jerichow, and A. Pfitzmann. Mixes in mobile communication systems: Location management with privacy. In *Information Hiding*, 1996.

[7] Foxs-News. Man accused of stalking ex-grilfriend with gps. *http://www.foxnews.com/story/0293313148700.*

[8] B. Gedik and L. Liu. A customizable k-anonymity model for protecting location privacy. In *ICDCS*, 2005.

[9] G. Ghinita, P. Kalnis, and S. Skiadopoulos. Prive: Anonymous location based queries in distributed mobile systems. In *WWW*, 2007.

[10] D. Goldberg, M. Reed, and P. Syverson. Onion routing for anonymous and private internet connections. In *CACM*, 1999.

[11] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys*, 2003.

[12] J. Hong and J. Landay. An architecture for privacy-sensitive ubiquitous computing. In *MobiSys*, 2004.

[13] H. Hu, J. Xu, and D. Lee. A generic framework for monitoring continuous spatial queries over moving objects. In *SIGMOD*, 2005.

[14] E. Hyytiä and J. Virtamo. Random waypoint model in cellular networks. *Wireless Network*, 2006.

[15] P. Kamat, W. Xu, W. Trappe, and Y. Zhang. Temporal privacy in wireless sensor networks. In *ICDCS*, 2007.

[16] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk. Enhancing source-location privacy in sensor network routing. In *ICDCS*, 2005.

[17] P. Karger and Y. Frankel. Security and privacy threats to its. In *World Congress on Intelligent Transport Systems*, 1995.

[18] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *ICPS*, 2005.

[19] D. Kifer and J. Gehrke. l-diversity: Privacy beyond k-anonymity. In *ICDE*, 2006.

[20] M. Kolahdouzan and C. Shahabi. Voronoi-based k nearest neighbor search for spatial network databases. In *VLDB*, 2004.

[21] J. Kong and X. Hong. Anodr: Anonymous on demand routing with untraceable routes for mobile adhoc networks. In *ACM MobiHoc*, 2003.

[22] W. Ku, R. Zimmermann, W. Peng, and S. Shroff. Privacy protected query processing on spatial networks. In *IEEE Workshop on Privacy Data Management*, 2007.

[23] F. Li, D. Cheng, M. Hadjieleftheriou, G. Kollios, and S. Teng. On trip planning queries in spatial databases. In *SSTD*, 2005.

[24] M. Mokbel, C. Chow, and W. Aref. The new casper: Query processing for location services without compromising privacy. In *VLDB*, 2006.

[25] K. Mouratidis, M. Yiu, D. Papadias, and N. Mamoulis. Continuous nearest neighbor monitoring in road networks. In *VLDB*, 2005.

[26] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao. Query processing in spatial network databases. In *VLDB*, 2003.

[27] USAToday. Authorities: Gps systems used to stalk woman. *http://www.usatoday.com/tech/news/2002-12-30-gps-stalker_x.htm.*

[28] T. Wang and L. Liu. Location privacy over road networks. *GIT-CC Technical Report*, 2009.

[29] R. Want, A. Hopper, V. Falco, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 1992.

[30] M. Yiu, C. Jensen, X. Huang, and H. Lu. Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In *ICDE*, 2008.