# *k-Trustee*: Location Injection Attack-resilient Anonymization for Location Privacy

Lei Jin, Chao Li, Balaji Palanisamy, and James Joshi
*School of Computing and Information, University of Pittsburgh, USA*
{lej17, chl205, bpalan, jjoshi}@pitt.edu

## Abstract

Cloaking-based location privacy preserving mechanisms have been widely adopted to protect users' location privacy when using location-based services. A fundamental limitation of such mechanisms is that users and their location information in the system are inherently trusted by the Anonymization Server without any verification. In this paper, we show that such an issue could lead to a new class of attacks called location injection attacks which can successfully violate users' in-distinguishability (guaranteed by *k-Anonymity*) among a set of users. We propose and characterize location injection attacks by presenting a set of attack models and quantify the costs associated with them. We then propose and evaluate *k-Trustee*, a trust-aware location cloaking mechanism that is resilient to location injection attacks and guarantees a lower bound on the user's in-distinguishability. *k-Trustee* guarantees that each user in a given cloaked region can achieve the required privacy level of *k-Anonymity* by including at least *k-1* other trusted users in the cloaked region. We demonstrate the effectiveness of *k-Trustee* through extensive experiments in a real-world geographic map and our experimental results show that the proposed cloaking algorithm guaranteeing *k-Trustee* is effective against various location injection attacks.

*Keywords:* Location privacy attack; Location privacy protection; Trust; k-Anonymity; k-Trustee.

## 1. Introduction

The rapid development of the high-speed mobile networks and the growing usage of the advanced mobile devices have made various location-based services to be indispensable in people's lives. Users' location privacy threats refer to the risks that an attacker can obtain unauthorized access to raw location data by locating a transmitting device and identifying the subject (person) using it. Examples of such risks include spamming users with unwanted advertisements, drawing sensitive inferences from victims' visits to various locations (e.g., clinics and doctors' offices) and learning sensitive information about them (e.g., diseases, religious and political affiliations, etc.). Hence, preserving location privacy is becoming a critical issue.

Various cloaking-based location privacy preserving mechanisms (CLPMs) have been proposed for protecting users' location privacy from location based service providers [4, 34, 20]. As shown in Figure 1, CLPMs are usually implemented through a trusted third party called Anonymization Server (AS) that collects users' location information and performs an anonymization prior to releasing the sensitive location information to location service providers (LBSPs), which are assumed to be either curious-but-honest or malicious. In some cases, the location service providers (LBSPs) are also vulnerable to insider threats. When a user $u$ with a mobile device requests a location-based service (e.g. searching for the nearest coffee shop) from an LBSP, the mobile user first sends the request including his exact location (e.g., longitude and latitude values) to AS. AS then runs a location cloaking algorithm to reduce the precision of $u$'s location to satisfy the required privacy level (e.g., *k-Anonymity*). After that, AS sends the cloaked region associated with $u$ to the LBSP which finally generates the answer to $u$'s request based on the information from AS. This answer is sent back to $u$ either directly, or through AS as an intermediate tier which delivers the answer to $u$ later.

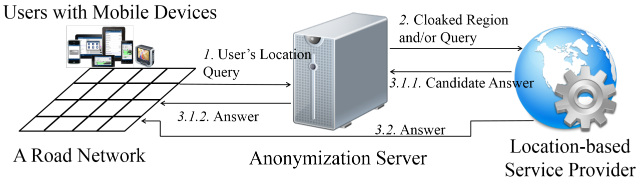


Figure 1: Architecture of a cloaking-based location privacy preserving mechanism.

One way to protect the location privacy of $u$ is to enhance the in-distinguishability of $u$ among a group of users, which is defined as *k-Anonymity* [4, 20]. Specifically, *k-Anonymity* guarantees that the location of a given user is indistinguishable from those of at least $k-1$ other users. In addition to *k-Anonymity*, several extended CLPMs have been proposed, such as POI (*points of interest*) *l-Diversity* [4], which ensures the in-distinguishability of a user's location from a set of POIs, and road segment *s-Diversity* [40], which guarantees the in-distinguishability of a user's

*Preprint submitted to Elsevier* *December 18, 2018*

location from a set of road segments. However, one fundamental limitation of these CLPMs is that all users and their location information have to be trusted by AS, which makes the CLPMs vulnerable in practice. Specifically, by exploiting this implicit assumption, an attacker can create fake users with carefully manipulated location information to forcibly reduce the privacy level guaranteed by CLPMs and significantly increase the chance of identifying a targeted user's location. Due to the limitation mentioned above, AS is unaware of the privacy level reduction caused by the injected fake users and therefore no precautionary measure or remedial measure can be implemented. In this paper, we first show that such vulnerability can lead to a new class of attacks called *location injection* attacks, which can successfully compromise privacy of the users' location and trajectory information. After characterizing the location injection attacks, we present various attack models and discuss the cost associated with them. Then, to mitigate the location injection attacks, we further propose a trust based mechanism called *k-Trustee*, which innovatively combines trust management with $k$-anonymity to distinguish fake users (untrusted users) from real users (trusted users) to make it resilient to the location injection attacks. The resilience of the proposed *k-Trustee* is theoretically analyzed and experimentally evaluated. In summary, the contributions of this paper are as follows:

- We first propose and characterize location injection attacks that can compromise users' privacy setting of *k-Anonymity* in an existing CLPM. We experimentally demonstrate the effectiveness of such attacks through simulations.

- Second, we propose the notion of trust in CLPMs and design a suite of trust-based location cloaking algorithms that can mitigate the impact of location injection attacks.

- Finally, we present the theoretical and experimental analyses of the proposed approaches to demonstrate and validate their effectiveness and resilience against location injection attacks.

The rest of the paper is organized as follows. In Section 2, we review the basic concepts of CLPMs. We then define the notion of location injection attacks in CLPMs and introduce the attack models. In Section 3, we define the concept of trust between users and introduce the notion of *k-Trustee* and design a cloaking algorithm that guarantees the *k-Trustee* property. In Section 4, we demonstrate the effectiveness of location injection attacks and experimentally evaluate the resilience of our proposed cloaking algorithms against location injection attacks. Finally, we summarize the related work in Section 5 and conclude the paper in Section 6.

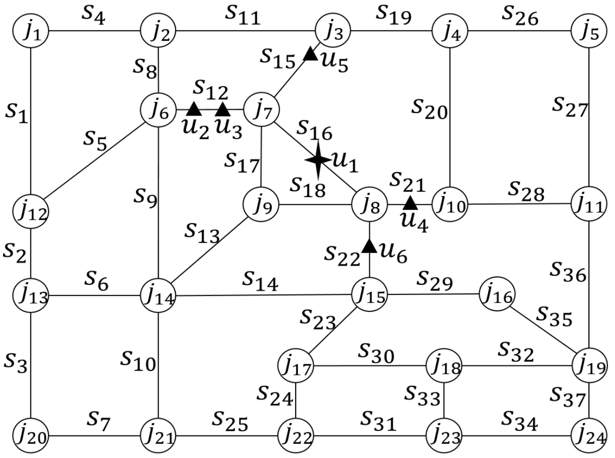


Figure 2: A road network example with 24 junctions and 37 road segments

## 2. Location Injection Attacks

In this section, we first model the road network and present the location cloaking techniques based on it. We then propose the location injection attacks and attack models.

### *2.1. Road Network Model*

In various cloaking approaches, users are assumed to travel in a road network [40] which is modeled as a graph $G(J, S)$, where $J$ represents the set of road junctions and $S$ represents the set of road segments. A junction is defined as the crossover point of any two roads or the end of a road segment. A road segment is defined as the direct road connecting any two adjacent junctions, which may include several point-of-interest (POI) venues. Each segment is uniquely determined by the two junctions associated with it while each junction is associated with one or more adjacent road segments. A road segment $s_i$ that connects two road junctions $j_p$, $j_q$ can be denoted by $s_i = (j_p, j_q)$. An example road network is shown in Figure 2 where there are 24 road junctions and 37 road segments.

In particular, for each road segment $s_i = (j_p, j_q)$ in the road network, we define the set of segments sharing either junction $j_p$ or junction $j_q$ with $s_i$ to be the neighbor set of $s_i$, which is denoted by $NS^{s_i}$. For example, in Figure 2, $NS^{s_1} = \{s_2, s_4, s_5\}$. Similarly, given a region $R$ including several road segments, $NS^R$ indicates the neighbor set of $R$, which consists of segments sharing at least one junction with the segments in $R$. In Figure 2, assuming that $R = \{s_1, s_2\}$, we have $NS^R = \{s_3, s_4, s_5, s_6\}$.

### *2.2. Location Cloaking Models*

In a road network, the objective of cloaking-based location privacy protection mechanisms (CLPM) is to preserve users' location privacy during their travels in the road network. The fundamental privacy notion behind conventional location cloaking models is *location k-Anonymity* [20, 17, 19], which guarantees the in-distinguishability of a user among a set of users. In other words, a user's location information exposed after the location cloaking process is indistinguishable from that of at least $k-1$ other users. Several extensions have also been proposed to enhance the privacy protection offered by cloaking based solutions, such as POI *l-Diversity* [4] which additionally ensures the in-distinguishability of a user's location from a set of POIs and road segment *s-Diversity* [40] which additionally guarantees the in-distinguishability of a user's location from a set of road segments. In this paper, we focus on the location cloaking models guaranteeing *k-Anonymity* and/or *s-Diversity.* We present the basic definitions below.

**Definition 1.** *k-Anonymity* [20, 17, 19]. A user $u$'s location is said to satisfy the *k-Anonymity* at time $t$, if there are at least $k-1$ other users present at the same cloaked region at $t$.

**Definition 2.** *s-Diversity* [40]. A user $u$'s location satisfies *s-Diversity* at time $t$, if there are at least $k-1$ other users at the same cloaked region at $t$ and there are at least $s$ road segments in the cloaked region.

Note that in this paper we set the atomic element of a cloaked region as a road segment in a road network [40]; *i.e.*, a cloaked region consists of only road segments. For example, in Figure 2, we assume that there are 6 users, $u_1$, $u_2$, $u_3$, $u_4$, $u_5$ and $u_6$, in the road network. We also assume that both the *k-Anonymity* and the *s-Diversity* requirements are 4 ($k = s = 4$) for $u_1$ and 3 ($k = s = 3$) for other users. In a CLPM that only guarantees the *k-Anonymity*, the cloaked region for $u_1$ can be the area consisting of $s_{12}$, $s_{15}$ and $s_{16}$. When the *s-Diversity* is supported by a CLPM, the cloaked region for $u_1$ can be the area composed of $s_{12}$, $s_{15}$, $s_{16}$ and $s_{17}$ as it ensures at least 4 segments in the cloaked region.

In the next section, we define the location injection attacks aiming to compromise the location privacy of mobile users, which work by manipulating locations of fake and/or compromised users[1]. We consider a user's location privacy is compromised when an attacker can either identify the road segment where the user is (e.g. find $s_{16}$ from $s_{12}$, $s_{15}$, $s_{16}$ for $u_1$ in Figure 2) or shrink the cloaked region to a smaller size (e.g. shrink $s_{12}$, $s_{15}$, $s_{16}$ to $s_{12}$, $s_{16}$ for $u_1$ in Figure 2), which breaches the user's privacy requirements (*k-Anonymity* and/or *s-Diversity*). We also refer to the violation of a user's trajectory privacy as the case where an attacker can identify a series of consecutive road segments a user visits.

### *2.3. Attack Definition*

To define the location injection attack, we assume, without loss of generality, that there is a road network $G(J, S)$, an attacker, a trusted user $u$ who travels in $G$ and requests a location-based service from an LBSP through an Anonymization Server (AS). The user $u$ has a privacy setting $k^u$ for *k-Anonymity* and AS guarantees the privacy requirement in the generated cloaked region using a cloaking-based location privacy preserving algorithm (e.g., PrivacyGrid [4], Casper [34], XStar [40]). We also assume that the attacker is the LBSP or a part of the LBSP that tries to compromise $u$'s privacy requirement of $k^u$, indicating a form of insider attack. The attacker (LBSP) knows the initial cloaked region including $u$ before launching the attacks from $u$'s recent location requests sent to the LBSP. Let a fake user be a user that does not physically exist but the attacker has created an account for him in the system, or an authentic user whose location can be manipulated by the attacker.

***Adversary's Action***: Let $u_i$ be a targeted user. An adversary's attack involves intelligently manipulating a number of fake users' locations using various schemes to identify $u_i$'s location. Let $R^{u_i}$ be the cloaked region created in response to a request from $u_i$. Let $U(R^{u_i})$ be the set containing all the users including $u_i$ in $R^{u_i}$ and $U_f(R^{u_i})$ be the set of fake users in $R^{u_i}$.

***Location Injection Attack***: We say that $u_i$ is a victim of a location injection attack, when $|U(R^{u_i})| - |U_f(R^{u_i})| < k^{u_i}$. Here $|U|$ indicates the number of users in a user set $U$.

In a location injection attack, an attacker can distinguish the fake users since these users are either created or controlled by the attacker. As a result, the number of remaining users in the cloaked region, namely $|U(R^{u_i})| - |U_f(R^{u_i})|$, becomes less than the user's privacy requirement of $k^{u_i}$. When this happens, a user's privacy requirement is compromised. In addition, the size of the cloaked region constructed for $u_i$ or the number of POIs in the cloaked region may also be controlled (e.g., decrease its size) by placing fake users in strategic locations. For example, as shown in Figure 3, there are six trusted users $u_1$, $u_2$, $u_3$, $u_4$, $u_5$ and $u_6$ traveling in a road network. An attacker can utilize six fake users $fu_1$, $fu_2$, $fu_3$, $fu_4$, $fu_5$ and $fu_6$ and report their locations in the road segments around the road junction, $Jun1$. We assume that $u_1$ has the *k-Anonymity* requirement of $k^{u_1} = 6$ and let the *k-Anonymity* requirements of other users be less than or equal to 6. Without the presence of fake users, AS may generate a cloaked region containing users $u_1$, $u_2$, $u_3$, $u_4$, $u_5$ and $u_6$. The probability of inferring $u_1$ from that of others in the cloaked region is 1/6. However, when the attacker launches a location injection attack, AS may generate a cloaked region including segments $Seg1$ and $Seg3$ where there are only two authentic users $\{u_1, u_2\}$ and four fake users $\{fu_2, fu_4, fu_5, fu_6\}$. Since the attacker can distinguish fake users in the constructed cloaked region, the probability of identifying $u_1$ from others is now reduced to 1/2, which compromises $u_1$'s privacy requirement of *k-Anonymity.* Hence, the attacker now has a higher probability of identifying $u_1$'s exact location; *i.e.*, $u_1$ could be

[1] In this paper, a compromised user refers to an authentic user whose location information can be arbitrarily manipulated by an attacker. In the rest of this paper, we simply use the notion of fake users to indicate the set of fake as well as compromised users utilized in location injection attacks.

traveling in $Seg1$, $Seg2$, $Seg3$, $Seg4$ or $Seg11$ without the attack but when the location injection attack is launched, $u_1$ would be associated with either $Seg1$ or $Seg3$.

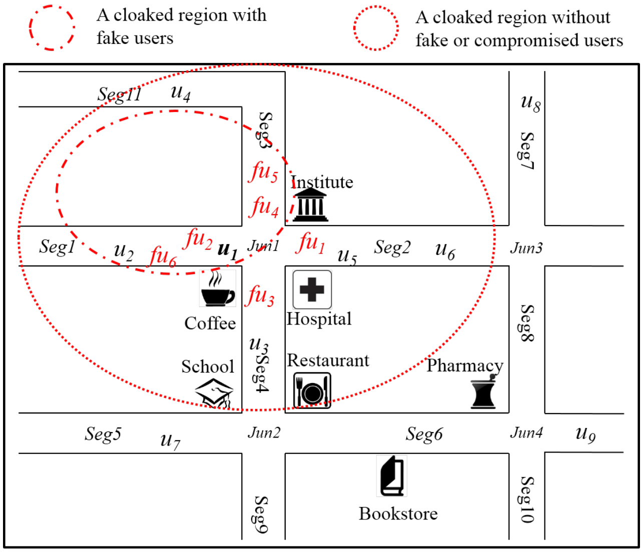


Figure 3: An instance of a location injection attack

Note that a location injection attack is successful only when the number of trusted users is less than that required to support a user's *k-Anonymity* requirements. For example, in Figure 3, if $k^{u_1} = 2$, then $u_1$'s privacy requirement is not violated even under the location injection attack. Detailed analysis of this attack model is provided in Section 3.4. In addition, a location injection attack can be targeted at multiple users simultaneously. It can be also used to infer a targeted user's trajectory when a sequence of location injection attacks for the targeted user are successful.

In our previous work [25], we simply defined the location injection attack where attackers can arbitrarily set their locations and thus their trajectories appear suspicious. In addition, we assume that fake users have the lowest *k-Anonymity* requirement supported by AS. This is because an attacker does not expect to trigger the expansion of a cloaked region and aims to induce the CLPM to construct a cloaked region as small as possible. In the next section, we model the location injection attacks.

### 2.4. Attack Models

We present the following three different location privacy attacks: *stalking attack*, *fixed-location attack* and *fixed-trajectory attack*. Generally speaking, location privacy attacks involve inferring a relationship between a user and his private location information based on the locations he has visited. Depending on the motive, an adversary may want to find out either '*the locations that have been visited by a targeted user*' or '*the users who have visited a chosen location of interest*'. In the first case, an adversary is more interested in learning private information about the user, so he can launch the stalking attack to continuously stalk the locations of that user and infer private information from collected locations. In the second case, an adversary may target a specific kind of private information (e.g., health information, political inclination) and be more interested in learning about the users associated with this private information. Here, the adversary can launch the fixed-location attack to continuously monitor the users visiting a specific location (e.g., a hospital) or the fixed-trajectory attack using a specific trajectory (e.g., a parade route) to monitor the users following that trajectory. These will help infer the relationship between the users and some private information. The main difference between the fixed-location attack and the fixed-trajectory attack is that in the first case private/sensitive information is implied by a visit to a specific location while in the second case private information is implied by a user's movement along a specific trajectory.

#### 2.4.1. Stalking Attack

When a location injection attack targets a specific user $u$, its main purpose is to compromise $u$'s privacy requirement for *k-Anonymity* and identify/infer more accurately his location at a specific time; e.g., the road segment where $u$ is located at time $t$. When the attacker has obtained a series of more accurate locations of $u$, he can infer or even identify the detailed trajectory of $u$. We call such an attack scenario *stalking* attack and we define it as follows.

***Assumption***: We assume that the attacker is the LBSP or a part of the LBSP that tries to compromise $u$'s privacy requirement of $k^u$, indicating a form of insider attack. Like many previous work [17, 29, 34], we assume that each LBS query contains a user ID (or pseudonym), so the attacker (LBSP) has the ability to track $u$'s cloaked regions. In addition, we assume that $u$ sends LBS queries with a high frequency, so that the attacker (LBSP) can frequently receive $u$'s cloaked regions and use them to stalk $u$. Finally, we assume the attacker (LBSP) can generate an arbitrary number of fake users to be located at any segment.

***Identifying Initial Road Segment of the Target***: To explain the identification of initial position of a user, we assume a user $u$ keeps sending LBS queries (with cloaked regions) to the attacker (LBSP) at $t = -1, 0, 1, 2...$ At $t = -1$, the attacker (LBSP) received $u$'s cloaked region $R^u_{-1}$, which contains no fake users. Then, between $t = -1$ and $t = 0$, the attacker (LBSP) decides to stalk $u$. For each road segment in $R^u_{-1}$ and its neighbor set $NS^{R^u_{-1}}$, the attacker places a number of fake users (e.g., the number of fake users deployed to each segment could be equal to the maximum value of $k$ that AS allows a user to declare, so $k \geq k^u$). All these fake users should periodically query the attacker (LBSP) through the anonymization server to be involved in $u$'s future cloaked regions. Later, when $t = 0$, $u$ sends the next LBS query to the attacker (LBSP) through the anonymization server, which will generate the next cloaked region containing $u$, denoted by $R^u_0$. We define the segment containing $u$ in $R^u_0$ as $u$'s initial segment, denoted by $s^u_{init}$. Since segment $s^u_{init}$ contains $k$ fake users and

$k \geq k^u$, $R_0^u$ will be $\{s_{init}^u\}$, so $s_{init}^u$ can be identified.

***Stalking Attack***: After $s_{init}^u$ has been identified, the attacker starts to stalk $u$. Specifically, the attacker makes the fake users move only within $NS^{s_{init}^u}$. Later, when $u$ moves to a new segment from $s_{init}^u$ and queries AS at $t = i$ ($i > 0$) to generate the cloaked region $R_i^u$, which is different from $R_0^u$, the attacker makes the fake users move into $R_i^u$ as soon as possible to identity $u$'s new position $s_i^u$ inside $R_i^u$. Similarly, when $u$ moves to another segment from $s_i^u$ and queries AS at $t = j$ ($j > i$) to generate the cloaked region $R_j^u$, the attacker tries to manipulate the trajectories of the fake users to identity $u$'s new position $s_j^u$ inside $R_j^u$. By repeating these steps, the attacker can keep stalking $u$.

***Example***: We show a comprehensive example of location injection attack in Figure 4, a part of the road network in Figure 2 . We assume the target user $u_1$ moves along the trajectory $s_{16} \rightarrow s_{12} \rightarrow s_9$. We assume $u_1$ sends three queries to AS at $t = -1, 0, 1$ when it moves along $s_{16}$, three queries to AS at $t = 2, 3, 4$ when it moves along $s_{12}$ and finally three queries to AS at $t = 5, 6, 7$ when it moves along $s_9$. We also assume the cloaked region at $t = -1$ is $R_{-1}^{u_1} = \{s_{16}, s_{21}\}$. At time $t = -1$, $R_{-1}^{u_1} = \{s_{16}, s_{21}\}$ is known by the attacker, so the attacker can place 5 fake users at each road segment within $R_{-1}^{u_1}$ and $NS^{R_{-1}^{u_1}}$. Then, at $t = 0$, since $R_0^{u_1} = \{s_{16}\}$, the initial segment is identified. Please notice that we only show the 5 fake users ($fu_1$, $fu_2$, $fu_3$, $fu_4$ and $fu_5$) assigned to segment $s_{16}$ and omit other fake users. After that, to stalk the user $u_1$, the attacker can dynamically create trajectories for the deployed fake users to make them always run after $u_1$. That is, given that the cloaked regions of $u_1$ at time $t = 2$ and $t = 5$ are $R_2^{u_1} = \{s_8, s_{11}, s_{12}, s_{15}\}$ and $R_5^{u_1} = \{s_9, s_{13}\}$, respectively, the attacker can control the deployed fakes users to enter the two regions to shrink them to $\{s_{12}\}$ and $\{s_9\}$, respectively, so that the trajectory $s_{16} \rightarrow s_{12} \rightarrow s_9$ can be disclosed.

### *2.4.2. Fixed-location Attack*

As another class of location injection attacks, an attacker may cast anchor at a specific location and aim to identify users who visit the targeted location, thus compromising the location privacy of the visitors. That is, instead of stalking a user to incrementally collect his sensitive locations, the attacker can select a fixed sensitive place (e.g., a hospital) and wait for the victims to fall into a snare. An attacker can manipulate the locations of the fake users to the targeted sensitive locations that are close to the targeted location. When users visit the targeted location, users' privacy requirements for the *k-Anonymity* are compromised with a higher probability. This is because the cloaked region has fake users who are controlled by the attacker. The probability of identifying a user is determined by the ratio of real users to fake users. To obtain a probability close to 100%, the adversary should estimate the number of real users and adjust the number of fake users based on that. We call such an attack *fixed-location* attack and it works as follows.

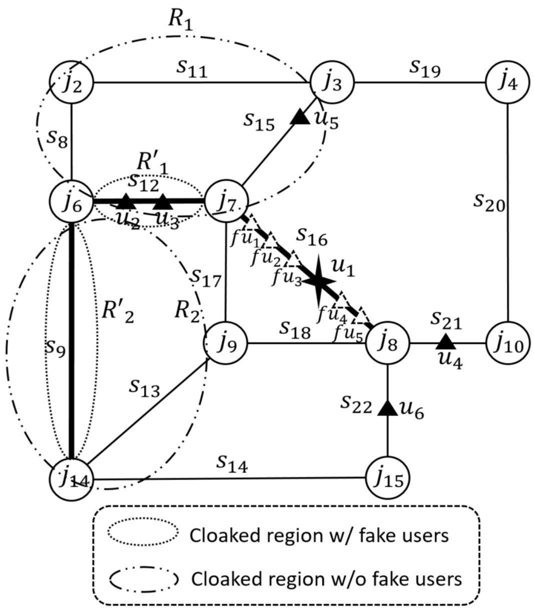


Figure 4: An example of the stalking attack. In the example, an adversity keeps using fakes users ($fu_1$, $fu_2$, $fu_3$, $fu_4$ and $fu_5$) to stalk the target user $u_1$ and successfully shrinks the cloaked regions $R_1$ and $R_2$ to smaller regions $R'_1$ and $R'_2$. Please notice that we only show the 5 fake users ($fu_1$, $fu_2$, $fu_3$, $fu_4$ and $fu_5$) assigned to segment $s_{16}$ and omit other fake users assigned to other segments.

***Assumptions***: The assumptions are same as the ones presented in Section 2.4.1.

***Fixed-location Attack***: The attacker places $m$ fake users at the targeted road segment $s$ and makes these fake users stay at $s$ (*e.g.*, visiting POIs at $s$) during the attack.

***Example***: In Figure 4, an attacker targets the road segment $s_{16}$ and tries to identify users who are traveling in $s_{16}$ by injecting $fu_1$, $fu_2$, $fu_3$, $fu_4$ and $fu_5$. We assume that a user $u_1$ is traveling in $s_{16}$ and she has a privacy setting $k^{u_1} = 4$. When $u_1$ requests a location-based service, AS selects $s_{16}$ as a cloaked region such that it satisfies $u_1$'s privacy requirement $k^{u_1}$. Since there is only $u_1$ in the cloaked region besides fake users, the attacker can determine $u_1$ is at $s_{16}$. Without launching such an attack, the cloaked region constructed for $u_1$ may consist of $s_{12}$, $s_{15}$ and $s_{16}$ which includes other three users who are not created by the attacker. In this case, the attacker cannot identify exactly where $u_1$ is located. It could be $s_{12}$, $s_{15}$ or $s_{16}$.

### *2.4.3. Fixed-trajectory Attack*

In certain situations, an attacker may be interested to identify users who travel at a specific trajectory consisting of a set of connected road segments. We call such an attack *fixed-trajectory* attack and it works as follows. (*e.g.*, $s_{16} \rightarrow s_{12} \rightarrow s_9$ in Figure 4).

***Assumptions***. The assumptions are same as the ones presented in Section 2.4.1.

***Fixed-trajectory Attack***. The attacker can first identify the smallest circular region that includes each road

| Notations | Descriptions |
|---|---|
| $u$ | real user. |
| $fu$ | fake user. |
| $s$ | road segment. |
| $NS^s$ | neighbor set of $s$. |
| $R_t^u$ | cloaked region of $u$ at $t$. |
| $t; \tau$ | time point; time duration. |
| $d_t^u(u_i, u_j)$ | coarse distance between $u_i$ and $u_j$ at $t$. |
| $d_t^s(u_i, s_j)$ | coarse distance between $u_i$ and $s_j$ at $t$. |
| $\sum_{t \in \tau} cut_t^u(u_i, u_j)$ | coarse-grained user-user trust during $\tau$. |
| $\sum_{t \in \tau} clt_t^s(u_i, s_j)$ | coarse-grained user-location trust during $\tau$. |
| $\sum_{t \in \tau} fut_t^u(u_i, u_j)$ | fine-grained user-user trust during $\tau$. |
| $\sum_{t \in \tau} flt_t^s(u_i, s_j)$ | fine-grained user-location trust during $\tau$. |
| $e^u$ | e-stalker parameter of $u$. |
| $f^u$ | f-stationary parameter $u$. |
| $(e_l^u, f_l^u)$ | local trust parameters of $u$. |
| $(e_g^u, f_g^u)$ | global trust parameters of $u$. |
| $U_{lt}^u(t)$ | local trustees of $u$. |
| $U_{gt}^u(t)$ | global trustees of $u$. |
| $U_T^u(t)$ | trustees of $u$. |
| $U_T^u(R_t^u)$ | trustees of $u$ in cloaked region $R_t^u$. |
| $k^u$ | $k$-Anonymity parameter of $u$. |
| $R_M^u$ | maximum acceptable cloaked region size of $u$. |
| $T_M^u$ | maximum acceptable response time of $u$. |
| $p^u$ | privacy parameters of $u$. |
| $\overline{U_{lt-e}^u(t)}$ | potential e-stalkers of $u$. |

Table 1: Summary of notations

segment in the targeted trajectory in the road network. Then, the attacker can simulate the trajectories of fake users continuously traveling in this circle. Note that the attacker has to create an adequate number of fake users at each road segment in the circle continuously in order to best induce AS to construct the cloaked regions that include only one road segment.

***Example***: In Figure 4, to identify the users travelling along the trajectory $s_{16} \rightarrow s_{12} \rightarrow s_9$, the attacker can simulate the trajectories of fake users continuously traveling in this circle $s_{16} \rightarrow s_{12} \rightarrow s_9 \rightarrow s_{13} \rightarrow s_{18} \rightarrow s_{16}$.

In summary, we can see that the stalking attack and the fixed-location attack mainly compromise users' location privacy while the fixed-trajectory attack can compromise users' trajectory privacy. In Section 4, we experimentally simulate these three attacks and demonstrate their effectiveness in successfully invading the location privacy of users. Next, we present the ways to mitigate these location injection attacks.

## 3. Mitigating Location Injection Attacks

In this section, we first discuss potential solutions to defend against location injection attacks. We then introduce various definitions related to trust computations and propose the trust based cloaking-based mechanism against location injection attacks, *k-Trustee*. Notations that will be used in this section are summarized in Table 1.

### *3.1. Discussions of Potential Solutions*

An intuitive approach to defend against location injection attacks is to design a detection mechanism for AS to detect fake users. When users are identified as fake by the detection process, their location-based requests can be rejected by AS. Such a detection approach can be based on the characteristics of a user (e.g., IP address) or the user's trajectory (*e.g.*, suspicious or abnormal trajectories). However, it has the following issues:

- It needs a verification process to validate the identified fake users, which incurs additional cost. It is also difficult to design such a process because of users' privacy preferences.
- There will always be false-positives and false-negatives in a detection approach. Trusted users will not be able to request any location-based service when they are identified as false-positives. For example, a trusted user may have a suspicious trajectory around a stadium while trying to find a parking slot (similar to the trajectories of fake users in the fixed-trajectory attack). A detection approach may mistakenly flag the trusted user as a fake user because of her suspicious trajectory. In addition, when the fake users are flagged as false-negatives, they can still be used in location injection attacks.
- It is also very difficult to completely characterize fake users and their suspicious trajectories.

We also note that the encryption-based approaches [11] to encrypt users' information and disconnect their identities with their locations are also feasible approaches to defend against the location injection attack. However, the cost of the encryption and decryption for each request of the location-based service from each user may be high, which makes such an approach less practical.

In this paper, we propose a trust based mitigation approach, named *k-Trustee*, that aims to reduce the impact of the location injection attacks through trust computations. Such a trust based mitigation approach has the following advantages:

- It does not detect nor validate fake users but it will mitigate the impact of suspicious users who could be either trusted users or fake users. Compared with detection approaches where false-positives and/or false-negatives are usually inevitable, the proposed mitigation approach will never forbid real users to request services.
- Users including fake users are always able to request services from AS and LBSPs. However, anonymity service is not free lunch. Users including attackers have to pay for that service. In this case, the attacker has to pay for a cost to conduct location injection attacks irrespective of whether the attacks are successful or not. Such a mechanism can significantly increase the attack cost for the attacker.

### 3.2. Trust Computations

In this subsection, we first introduce the computations of the user-user trust and the user-location trust and then apply these trust computations to define *k-Trustee.*

#### 3.2.1. Trust Functions

The principle behind the computation of trust is that a user $u_j$ is more trusted by another user $u_i$ or a road segment $s_i$ if $u_j$ is always further away from $u_i$ or $s_i$. When $u_j$ follows $u_i$ ($u_j$ is always close to $u_i$) or $u_j$ is always traveling around $s_i$, we say that $u_j$ has a probability to be a fake/compromised user targeting $u_i$ or $s_i$ in the attack. Thus, $u_j$ may not be trusted by $u_i$ or $s_i$. We first define two types of distances.

**Definition 3.** *User-User Distance*. Given a road network $G(J, S)$ and two users $u_i$ and $u_j$, we use $d^u_t(u_i, u_j)$ to represent the coarse distance between $u_i$ and $u_j$ at time $t$. When $u_i$ and $u_j$ appear together in a same cloaked region $R$, $d^u_t(u_i, u_j) = 0$. In other cases, $d^u_t(u_i, u_j) = SJ(u_i, u_j)$. Here, $SJ(u_i, u_j)$ is equal to the number of the junctions in the shortest path between the locations of $u_i$ and $u_j$ in a road network.

**Definition 4.** *User-Location Distance*. Given a road network $G(J, S)$, a user $u_i$ located at a road segment $s_i$, and a road segment $s_j$, we use $d^s_t$ to represent the coarse distance between $u_i$ and $s_j$ at time $t$. When $u_i$ is in a cloaked region $R$ including $s_j$, $d^s_t(u_i, s_j) = 0$. Otherwise, $d^s_t(u_i, s_j) = SS(s_i, s_j)$, where $SS(s_i, s_j)$ is equal to the number of junctions in the shortest path between $s_i$ and $s_j$.

For example, in Figure 3, $d^u_t(u_1, fu_4) = 0$ since $u_1$ and $fu_4$ are in the same cloaked region. $d^u_t(u_1, u_9) = 3$ as there are three junctions in the shortest path between $u_1$ and $u_9$. Similarly, $d^s_t(u_1, Seg_3) = 0$ since the cloaked region includes both $u_1$ and $Seg_3$; $d^s_t(u_1, Seg_9) = 1$ because there is one junction in the shortest path between $u_1$ and $Seg_9$.

Based on the definitions 3 and 4, we then present two types of user-user trust functions and two types of user-location trust functions. The user-user trust functions present the trust between users while the user-location trust functions indicate the trust values from locations to users.

**Definition 5.** *Coarse-grained User-User Trust Function*. Given a road network $G(J, S)$, two users $u_i$ and $u_j$ traveling in $G$ and a time interval $\tau$, the coarse-grained user trust between $u_i$ and $u_j$ is $\sum\limits_{t\in\tau} cut^u_t(u_i, u_j)$. Here,

$$cut^u_t(u_i, u_j) = \begin{cases} 1, d^u_t(u_i, u_j) = 0 \\ 0, d^u_t(u_i, u_j) > 0 \end{cases}$$

**Definition 6.** *Coarse-grained User-Location Trust Function*. Given a road network $G(J, S)$, a user $u_i$ traveling in $G$, a road segment $s_j$ ($s_j \in J$) and a time interval $\tau$, the coarse-grained location trust function between $u_i$ and $s_j$ is $\sum\limits_{t\in\tau} clt^s_t(u_i, s_j)$. Here,

$$clt^s_t(u_i, s_j) = \begin{cases} 1, d^s_t(u_i, s_j) = 0 \\ 0, d^s_t(u_i, s_j) > 0 \end{cases}$$

**Definition 7.** *Fine-grained User-User Trust Function*. Given a road network $G(J, S)$, two users $u_i$ and $u_j$ traveling in $G$ and a time interval $\tau$, the fine-grained user trust between $u_i$ and $u_j$ is $\sum\limits_{t\in\tau} fut^u_t(u_i, u_j)$, where

$$fut^u_t(u_i, u_j) = \begin{cases} 1, d^u_t(u_i, u_j) = 0 \\ xd^u_t(u_i, u_j)^{-y}, d^u_t(u_i, u_j) > 0 \end{cases}$$

Here, $x \geq 0, y \geq 0, 0 < xd^u_t(u_i, s_j)^{-y} < 1$.

**Definition 8.** *Fine-grained User-Location Trust Function*. Given a road network $G(J, S)$, a user $u_i$ traveling in $G$, and a road segment $s_j$ ($s_j \in J$) and a time interval $\tau$, the coarse-grained location trust function between $u_i$ and $s_j$ is $\sum\limits_{t\in\tau} flt^s_t(u_i, s_j)$, where

$$flt^s_t(u_i, s_j) = \begin{cases} 1, d^s_t(u_i, s_j) = 0 \\ xd^s_t(u_i, s_j)^{-y}, d^s_t(u_i, s_j) > 0 \end{cases}$$

Here, $x \geq 0, y \geq 0, 0 < xd^s_t(u_i, s_j)^{-y} < 1$.

In both definition 7 and 8, users outside the cloaked regions are not simply considered to be innocent. Their degree of suspicion can be controlled by adjusting the parameters $x$ and $y$. Specifically, by choosing a larger $x$ while a smaller $y$, the user $u_i$ outside the cloaked region containing $u_j$ or $s_j$ obtains higher $fut^u_t(u_i, u_j)$ or $flt^s_t(u_i, s_j)$, thus becoming more suspicious. In contrast, by choosing a smaller $x$ while a larger $y$, users outside the cloaked regions become less suspicious. In this paper, we set $x = 1$ and $y = 2$, which considers users closer to the cloaked regions to be suspicious but their degree of suspicion is much lower than that of the users inside the cloaked region. Note that the time window $\tau$ in the definitions 5 - 8 is generally defined by AS. The value is same for all users. An example of such a time window could be 24 hours.

Based on definitions 5 and 7, when $u_i$ and $u_j$ are included in the same cloaked region or they are close to each other, the values of $\sum\limits_{t\in\tau} cut^u_t(u_i, u_j)$ and $\sum\limits_{t\in\tau} fut^u_t(u_i, u_j)$ are higher. The smaller distance between $u_i$ and $u_j$ also implies that $u_i$ may stalk $u_j$ or vice versa. Hence, the higher values of $\sum\limits_{t\in\tau} cut^u_t(u_i, u_j)$ and $\sum\limits_{t\in\tau} fut^u_t(u_i, u_j)$ refer to the lower trust between $u_i$ and $u_j$. Similarly, in the definitions 6 and 8, the higher values of $\sum\limits_{t\in\tau} clt^s_t(u_i, s_j)$ and $\sum\limits_{t\in\tau} flt^s_t(u_i, s_j)$ refer to the smaller distance between $u_i$ and $s_j$ and this suggests the lower trust from $s_j$ to $u_i$. In addition, compared to the coarse-grained trust functions (definitions 5 and 6), the fine-grained trust functions (definitions 7 and 8) are more restricted; users are probably regarded as potential attacker nodes even when they are just a bit close to a target but they are not included in the same cloaked region with the target. Instinctively, these fine-grained trust functions would be more effective to defend against the attacks, and they are more useful for handling the attack scenarios where fake users are placed a bit far away from a target for the attacks. However, the potential issue with the fine-grained trust functions is that they may consider more trusted users as suspicious users than the coarse-grained trust functions. Such an issue may make AS construct a larger size of a cloaked region and it may lower the quality of the location based services for users. We compare these two types of trust functions in Section 4.

Next, based on the above trust functions, we introduce the definitions of the local trust and the global trust which are used to capture the trust values between users and between users and road segments in a more comprehensive way. Based on these, we define the *k-Trustee.*

### *3.2.2. k-Trustee*

In order to define the local trust and the global trust, we first define the notions of *e-stalker* and *f-stationary* based on the proposed trust functions.

**Definition 9.** *e-stalker.* Given a road network $G(J, S)$, two users $u_i$ and $u_j$ traveling in $G$ and a time interval $\tau$, we say that $u_j$ is an *e-stalker* for $u_i$ when $\sum_{t\in\tau} cut^u_t(u_i, u_j) \geq e^{u_i}_l$ or $\sum_{t\in\tau} fut^u_t(u_i, u_j) \geq e^{u_i}_l$ . Here, $e^{u_i}_l$ is a parameter defined by $u_i$ indicating his privacy setting for *e-stalker.*

**Definition 10.** *f-stationary.* Given a road network $G(J, S)$, a user $u_i$ located at a road segment $s_i$, another user $u_j$ located at a road segment $s_j$, and a time interval $\tau$, we say that $u_j$ is an *f-stationary* of $u_i$ when $\sum_{t\in\tau} clt^s_t(u_j, s_i) \geq f^{u_i}_l$ or $\sum_{t\in\tau} flt^s_t(u_j, s_i) \geq f^{u_i}_l$. Here, $f^{u_i}_l$ is defined by $u_i$ specifying the privacy setting for *f-stationary.*

From these two definitions, we can see that *e-stalker* characterizes users who may be utilized by an attacker to identify and/or infer a specific user's location while *f-stationary* characterizes users who may be employed by the attacker to identify users who are visiting a specific location. Next, we define the *Local Trust* specifying whether a user trusts another locally.

**Definition 11.** *Local Trust.* Given two users $u_i$ and $u_j$ traveling in a road network $G(J, S)$, a time window $\tau$, $u_i$'s location $s_i$ at time $t$ ($t \in \tau$), and $u_i$'s local trust parameters $e^{u_i}_l$ and $f^{u_i}_l$, we say that $u_j$ is currently a local trusted user of $u_i$, denoted as $u_j \in U^{u_i}_{lt}(t)$, only when $u_j$ is neither an *e-stalker* nor an *f-stationary* of $u_i$. That is, $\sum_{t\in\tau} cut^u_t(u_i, u_j) < e^{u_i}_l$ and $\sum_{t\in\tau} clt^s_t(u_j, s_i) < f^{u_i}_l$, or $\sum_{t\in\tau} fut^u_t(u_i, u_j) < e^{u_i}_l$ and $\sum_{t\in\tau} flt^s_t(u_j, s_i) < f^{u_i}_l$.

Fake users used for a particular target can be reused by an attacker to attack a new target; these fake users may be initially trusted by the new target. To limit re-usability of fake users, we present the notion of *global trust* as follows.

**Definition 12.** *Global Trust.* Given two users $u_i$ and $u_j$ traveling in a road network $G(J, S)$, a time window $\tau$, $u_i$'s location $s_i$ at time $t$ ($t \in \tau$), and $u_i$'s global trust parameters $e^{u_i}_g$ and $f^{u_i}_g$, we say that $u_j$ is globally trusted by $u_i$, denoted as $u_j \in U^{u_i}_{gt}(t)$, only when there are less than $e^{u_i}_g$ users who regard $u_j$ as the *e-stalker*, and less than $f^{u_i}_g$ users who regard $u_j$ as the *the f-stationary.*

Now, when a fake user has been adopted for attacking enough users and/or road segments in the past, he is unlikely to be globally trusted by many other users. Hence, the re-usability of this fake user for a new target will be restricted.

We define a trusted user of a specific user by considering both local and global trust as follows.

**Definition 13.** *A Trustee of a Specific User.* Given two users $u_i$ and $u_j$ traveling in a road network $G(J, S)$, a time window $\tau$, $u_i$'s location $s_i$ at time $t$ ($t \in \tau$), $u_i$'s local trust parameters $e^{u_i}_l$ and $f^{u_i}_l$, and $u_i$'s global trust parameters $e^{u_i}_g$ and $f^{u_i}_g$, we say that $u_j$ is a trustee of $u_i$, denoted as $u_j \in U^{u_i}_T(t)$, only when $u_j \in U^{u_i}_{lt}(t)$ and $u_j \in U^{u_i}_{gt}(t)$.

In the rest of the paper, when we say $u_j$ is trusted by $u_i$, it will refer to local and global trust. We next present the notion of *k-Trustee* for a user as follows.

**Definition 14.** *k-Trustee of a User.* Given a road network $G(J, S)$, an Anonymization Server (AS) and a time window $\tau$, a user $u_i$ travels in $G$ while requesting a location-based service. $u_i$ has a privacy setting $k^{u_i}$ for *k-Anonymity* and AS constructs a cloaked region $R^{u_i}_t$ for $u_i$ at time $t$ ($t \in \tau$). $U^{u_i}_T(R^{u_i}_t)$ represents the trusted users of $u_i$ in the cloaked region $R^{u_i}_t$ at $t$. We say that *k-Trustee* is guaranteed for $u_i$ if and only if there are at least $k^{u_i}$ users in $U^{u_i}_T(R^{u_i}_t)$; *i.e.*, $|U^{u_i}_T(R^{u_i}_t)| \geq k^{u_i}$.

Note that we assume that $u_i$ always trusts himself (i.e., $u_i \in U^{u_i}_T(R^{u_i}_t)$).

The trustees of a specific user $u_i$ are the least likely to be fake users since these users have the lowest probability of either stalking $u_i$ or attacking the location where $u_i$ is currently is. When there are at least $k$ trustees in a cloaked region for user $u_i$, the probability of distinguishing $u_i$ in the cloaked region is at most $1/k^{u_i}$. Thus, $u_i$'s privacy requirement for *k-Anonymity* is guaranteed. Note that it is possible that fake users may be identified as trustees of a specific user $u_i$ in the initial stages. However, when these fake users continue to stalk $u_i$ or attack road segments including $u_i$, their trust values with respect to $u_i$ will keep decreasing, as per the proposed definitions. Eventually, they will not become the trustees of $u_i$ any more in the time window $\tau$. In this case, an attacker has to use new fake users to launch the location injection attacks on $u_i$. In addition, it is also possible that an authentic and not compromised user may not be always identified as a trustee of any user in terms of his trajectory. It is a false-negative but there is no impact for this authentic user to request anonymity service from AS and various location-based services from LBSPs. The only potential issue is that AS may construct a larger cloaked region for the authentic user and hence the quality of the location-based service may decrease.

Based on the definition of *k-Trustee* of a user, we define the notion of *guarantee of k-Trustee* as follows.

**Definition 15.** *Guarantee of k-Trustee in a Cloaked Region.* Given a cloaked region $R$ and a time instant $t$, a user set $U(R)$ indicates a set of users in $R$. We say that *k-Trustee* is guaranteed in $R$ at $t$ if and only if *k-Trustee* is guaranteed for each user in $R$ at $t$; *i.e.*, $\forall u_i \in U(R), |U_T^{u_i}(R_t^{u_i}, t)| \geq k^{u_i}$.

Next, we present the cloaking-based location privacy mechanism which guarantees *k-Trustee* in any cloaked region constructed by AS.

### *3.3. Cloaking-based Location Privacy Mechanism Guaranteeing k-Trustee*

Here, we first present the proposed *k-Trustee* cloaking based privacy framework. We then discuss and compare several expansion schemes and finally show the *k-Trustee* cloaking algorithm.

#### *3.3.1. k-Trustee Framework*

The key idea of our proposed cloaking-based location privacy framework is to adopt the notion of *k-Trustee* instead of the *k-Anonymity* to enhance a user's location privacy and mitigate the location injection attacks. We also suggest to place a sensor agent at each road segment of a road network in the original location privacy preserving framework shown in Figure 1. The sensor agent at a road segment is able to communicate with AS and the users close to the road segment. It is used to compute *f-stationary* values for users. After that, it sends the related *f-stationary* values to AS. Note that the sensor agents are not necessary if the computations of the *f-stationary* values are done by users' mobile devices. To do this, user $u_i$ should use cloaked regions previously received from AS within the past time period $\tau$. For any user that appeared at least once in these recent cloaked regions, user $u_i$ should count the number of times that this user appeared at each segment during $\tau$. The results can be used in $\sum_{t\in\tau} clt_t^s(u_j, s_i) < f_l^{u_i}$ to compute local *f-stationary* (Definition 10, 11). Finally, the local *f-stationary* can be sent to AS to compute the global *f-stationary* (Definition 12). We note that the *f-stationary* values computed by mobile devices may be less accurate than the ones computed by the sensor agents because the cloaked regions owned by mobile devices may not contain enough information about the surrounding users. Ideally, a sensor agent placed on the road segments can provide precise information about the users. However, it may not be entirely practical. In such cases, when users are away from the sensor agents, the *f-stationary* values can be computed by their mobile devices and when users are close to the sensor agents, the *f-stationary* values can be computed by the sensor agents. For example, when there are ten segments and ten other users appeared at least once in $u_i$'s recent cloaked regions and only two of the segments have sensor agents, $u_i$ can use mobile devices to compute a portion of local *f-stationary* values by counting the number of times that each of the ten users appeared at each of the eight no-sensor segment and importing the results into $\sum_{t\in\tau} clt_t^s(u_j, s_i) < f_l^{u_i}$. Later, at AS, this portion of local *f-stationary* values can be combined with the *f-stationary* values of the two with-sensor segments reported by the sensor agents to offer complete *f-stationary* information for $u$.

In this framework, a user $u$ first needs to specify his privacy requirement as a 7-tuple $p^u(k^u, e_l^u, f_l^u, e_g^u, f_g^u, R_M^u, T_M^u)$. Here, $R_M^u$ denotes the maximum size of a constructed cloaked region accepted by $u$; and $T_M^u$ indicates the maximum wait time accepted by $u$ for the response for his location request. $R_M^u$ and $T_M^u$ are usually used by $u$ to specify the quality of service. In this paper, $R_M^u$ refers to the maximum number of road segments in a road network. When the number of road segments in the cloaked region is larger than $R_M^u$, $u$'s location request will be ignored. $p^u$ needs to be sent to AS and sensor agents at road segments before the anonymous service is provided.

The process to compute trustees involves the following steps. First, *e-stalkers* are computed and labeled by the users. To compute *e-stalkers*, user $u_i$ should use cloaked regions previously received from AS within the past time period $\tau$. Specifically, for a user $u_j$ that appeared at least once in these recent cloaked regions, user $u_i$ should count the number of the recent cloaked regions that contains $u_j$. If the result is not less than $e_l^{u_i}$, $u_j$ will be labeled as a *e-stalker* of $u_i$ by $u_i$. To query AS for a cloaked region, $u_i$ should compute his *e-stalkers* and send the results to AS along with his current location $s_t^{u_i}$ and privacy parameters $p^{u_i}$. As a result, AS knows the *e-stalkers* of each user. Second, the computation of *f-stationary* is completed by sensor agents and AS together. We have assumed that each road segment has a sensor agent to continuously measure the *f-stationary* value $\sum_{t\in\tau} clt_t^s(u, s)$ for nearby users. For instance, once a user $u_j$ enters the communication area of the sensor agent located at segment $s_i$, the value $\sum_{t\in\tau} clt_t^s(u_j, s_i)$ will be continuously updated by the sensor agent until the value becomes zero. Therefore, if AS needs to determine whether or not user $u_j$ currently located at $s_t^{u_i} = s_i$ is a *f-stationary* of user $u_i$, AS should get the value $\sum_{t\in\tau} clt_t^s(u_j, s_i)$ from the sensor agent of $s_i$ and compare it with the $f_l^{u_i}$ declared by $u_i$. If the result is not less than $f_l^{u_i}$, $u_j$ will be labeled as a *f-stationary* of $u_i$ by AS. With the knowledge of both *e-stalker* and *f-stationary*, AS can label a user $u_j$ to be locally trusted by another user $u_i$ if $u_j$ is neither an *e-stalker* nor a *f-stationary* of $u_i$. In addition, with the global knowledge about the number of times that $u_j$ has been labeled as *e-stalker* and/or *f-stationary* by other users, AS can compare the results with parameters $e_g^{u_i}$ and $f_g^{u_i}$ declared by $u_i$ to determine whether $u_j$ can be globally trusted by $u_i$ or not. Finally, if $u_j$ can be both locally and globally trusted by $u_i$, AS will label $u_j$ to be a trustee of $u_i$.

An essential issue in this framework is the strategies for expanding a cloaked region where *k-Trustee* is guaranteed for each user. Generally, there are two approaches for expanding the cloaked region [4] in the literature: the *Bottom-Up* cloaking and the *Top-Down* cloaking. The *Bottom-Up* cloaking approach starts the cloaking process by taking a road segment as a candidate cloaked region. If it cannot satisfy users' privacy requirements, the *Bottom-Up* cloaking approach will start the expansion process to enlarge it by including more neighboring road segments till all the users' privacy requirements in the cloaked region are satisfied. On the other hand, the *Top-Down* cloaking approach first selects the entire graph as an initial candidate cloaked region and it aims to partition it to various smaller cloaked regions where none of the users' privacy requirements is violated. In this paper, we focus on the *Bottom-Up* cloaking approach since we feel that it is more straightforward and easier to be utilized in a road network.

### *3.3.2. Expansion Schemes*

We proposed the following three cloaked region expansion schemes in the proposed *k-Trustee* framework: *random* expansion, *greedy* expansion and *hybrid* expansion. Note that these are deployed in AS.

***Random Expansion***. Given a cloaked region $R$, the random expansion approach randomly picks a road segment from the neighbor set $NS^R$ and adds it to $R$. This process is repeated until the expanded $R$ guarantees the *k-Trustee* requirement for each user in $R$.

***Greedy Expansion***. The greedy expansion focuses on constructing a smaller size of the cloaked region at each expansion step. Given a cloaked region $R$, it first computes the neighbor set $NS^R$ when $R$ does not satisfy the *k-Trustee* requirement of each and every user. After that,in each expansion step, it tries to find the best road segment in $NS^R$ that can satisfy the users' privacy requirements as soon as possible. It then adds the road segment to $R$. It keeps adding the best road segment to $R$ until every user's requirement of *k-Trustee* is guaranteed. Below, we define the approach to identify the best road segment to add at each step, as follows:

**Definition 16.** *"Best First"*. Given a road network $G(J, S)$ and a cloaked region $R$ where not all users' *k-Trustee* privacy requirements are satisfied, let $NS^R$ be the neighbor set at time $t$. For each road segment $s_i \in NS^R$, let $p(s_i)$ be a profit function and let $c(s_i)$ be a cost function. $p(s_i)$ denotes the number of pairs of trusted users between a user in $s_i$ and another user in $R$. It can be calculated as $p(s_i) = \sum Tr(u_i, u_j), u_i \in U(s_i), u_j \in U(R)$, where

$$Tr(u_i, u_j) = \begin{cases} 1/(k^{u_j} - 1), u_i \in U_T^{u_j}(R \cup s_i, t) \\ 0, otherwise \end{cases}$$

$c(s_i)$ indicates the number of additional trusted users required for users at $s_i$ when $s_i$ is added to $R$. $c(s_i)$ can be computed as $c(s_i) = \sum_{u_i \in U(s_i)} \frac{(k^{u_i}-1) - \left|U_T^{u_i}(R \cup s_i)\right|}{k^{u_i}-1}$. We say that $s_i$ is the best road segment to add to $R$ when the value of $p(s_i) - c(s_i)$ is the largest, compared to other road segments in $NS^R$. When there are more than one best road segments, we randomly pick one of them and add it to $R$.

**Example:** We illustrate the working of the above expansion scheme as follows. Figure 5 shows the steps of a greedy expansion for $u_1$. $fu_1$, $fu_2$ and $fu_3$ in the figure are fake users. We first assume that every fake user has a very low privacy requirement ($k^{fu_1} = k^{fu_2} = k^{fu_3} = 2$) and trust any other user locally and globally in order to achieve the best attack results. Authentic users ($u_i, i \in [1, 12]$) do not trust these fake users globally but they trust any other authentic user globally; *i.e.*, $U_{gt}^{u_i} = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9, u_{10}, u_{11}, u_{12}\}, i \in [1, 12]$. Authentic users have privacy settings for the *k-Trustee* as: $k^{u_3} = k^{u_4} = k^{u_6} = k^{u_7} = k^{u_9} = k^{u_{10}} = k^{u_{11}} = 3$, $k^{u_1} = k^{u_2} = k^{u_5} = 4$, $k^{u_8} = k^{u_{12}} = 5$. Their current trustees are: $U_{lt}^{u_1} = U_{lt}^{u_9} = U_{lt}^{u_{10}} = \{u_4, u_5, u_6, u_7, u_8\}$, $U_{lt}^{u_2} = \{u_6, u_7, u_{10}\}$, $U_{lt}^{u_3} = \{u_5, u_7, u_{12}\}$, $U_{lt}^{u_4} = \{u_5, u_6\}$, $U_{lt}^{u_5} = \{u_1, u_4, u_6\}$, $U_{lt}^{u_6} = \{u_1, u_4, u_5, u_7, u_9\}$, $U_{lt}^{u_7} = \{u_4, u_6, u_8\}$, $U_{lt}^{u_8} = \{u_1, u_2, u_4, u_5, u_6\}$, $U_{lt}^{u_{11}} = \{u_2, u_5, u_6, u_7, u_8\}$, $U_{lt}^{u_{12}} = \{u_2, u_5, u_6, u_7, u_9\}$.

To construct a cloaked region for $u_1$, initially, in Figure 5, AS sets the candidate cloaked region $R^{u_1} = \{s_{16}\}$ and $NS^{R^{u_1}} = \{s_{12}, s_{15}, s_{17}, s_{18}, s_{21}, s_{22}\}$. Since $k^{u_1} = 4$ and there are less than 4 trustees in $R^{u_1}$, AS needs to expand $R^{u_1}$ by adding the best road segment in $NS^{R^{u_1}}$. Given $s_{12}$, $U(s_{12}) = \{u_8, u_9\}$, $p(s_1) = 1/3$ since only $u_8$ is one of the trusted users of $u_1$. $c(s_1) = 3/4 + 1/2 = 5/4$ as $u_8$ needs 3 more trusted users and $u_9$ needs 1 more trusted user. $p(s_{12}) - c(s_{12}) = -11/12$. Similarly, $p(s_{15}) - c(s_{15}) = 0 - 2 = -2$, $p(s_{17}) - c(s_{17}) = 0 - 1 = -1$, $p(s_{18}) - c(s_{18}) = 0 - 2 = -2$, $p(s_{21}) - c(s_{21}) = 1/3 - 1 = -2/3$, $p(s_{22}) - c(s_{22}) = 1 - 0 = 1$. Hence, $s_{22}$ is selected according to definition 16 and $CR^{u_1} = \{s_{16}, s_{22}\}$. We also find that every user's privacy requirement is satisfied in $R^{u_1}$ now and hence $R^{u_1} = \{s_{16}, s_{22}\}$ is selected as the cloaked region for $u_1$.

We can see that the greedy expansion tends to minimize the size of a cloaked region while the random expansion may generate a cloaked region with a larger size which can decrease the quality of the location-based services (QoS). However, the greedy expansion may be more vulnerable to the replay attack [40] where an attacker knows the preference of the expansion process and he can possibly replay the anonymization process to identify a

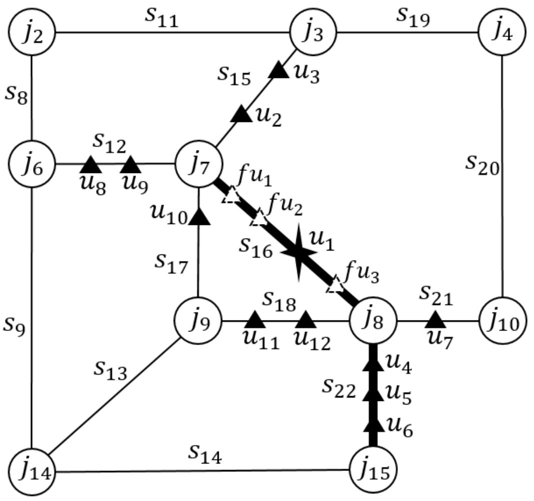


Figure 5: An example of the greedy expansion. In this example, $u_1$ requires a cloaked region that should satisfy $k^{u_1} = 4$. Initially, the candidate cloaked region $R^{u_1} = \{s_{16}\}$ and $NS^{R^{u_1}} = \{s_{12}, s_{15}, s_{17}, s_{18}, s_{21}, s_{22}\}$. To expand the cloaked region to satisfy $k^{u_1} = 4$, the greedy expansion computes the difference between profit and cost when each segment in $NS^{R^{u_1}}$ is added into $R^{u_1}$ and selects the segment $s_{22}$ with the largest difference.

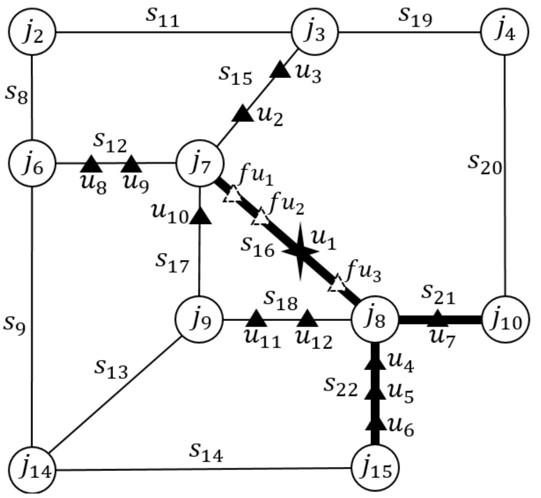


Figure 6: An example of the hybrid expansion. In this example, to expand the cloaked region $R^{u_1} = \{s_{16}\}$ to satisfy $k^{u_1} = 4$, in the first step, the hybrid expansion randomly selects the random expansion and therefore randomly selecting $s_{21}$ from $NS^{R^{u_1}}$. Then, since $k^{u_1} = 4$ is still not met, in the second step, the hybrid expansion randomly selects the greedy expansion and therefore selecting $s_{22}$ according to the 'best first' computation.

user's exact location. To balance the QoS and the resilience against the replay attack, we further propose a hybrid expansion which combines the random expansion and the greedy expansion.

***Hybrid Expansion***. When a cloaked region $R$ needs to be expanded, AS randomly adopts either the random expansion or the greedy one to add a road segment to $R$ in the hybrid expansion scheme. AS continues to do the same expansions until every user's privacy requirement is satisfied in $R$.

For example, in Figure 6, every user's privacy setting is same as the one in Figure 5. Initially, $R^{u_1} = \{s_{16}\}$ and $NS^{R^{u_1}} = \{s_{12}, s_{15}, s_{17}, s_{18}, s_{21}, s_{22}\}$. In the first expansion, we assume that the random expansion is employed and $s_{21}$ is added into $R^{u_1}$, so $R^{u_1} = \{s_{16}, s_{21}\}$. In the second expansion, the greedy expansion is adopted. $s_{22}$ is chosen based on definition 16 and it is added to $R^{u_1}$. Now, we can see that *k-Trustee* is guaranteed for every user in $R^{u_1} = \{s_{16}, s_{21}, s_{22}\}$.

### *3.3.3. k-Trustee Cloaking Algorithm*

The cloaking algorithm that guarantees *k-Trustee* is shown in Algorithm 1. In this algorithm, the cloaking process, run by AS, first initializes a user set $CU$ indicating users who have been processed by the algorithm, the output set and the expansion scheme adopted by the algorithm (line 1-3). It then computes the trusted users for each user at $t$ based on the user's privacy setting $p^{u_i}$ (line 4-5). After that, it randomly selects one user who has not been processed and starts to construct a cloaked region where each user's privacy requirement is satisfied using the selected expansion scheme (line 6-12). If the size of the cloaked region is larger than the required one for a user in the constructed cloaked region, the anonymity service is not available for that user (line 13-16). However, the AS will continuously include that user into future cloaked regions until the maximum response time $R_M^u$ of that user has passed. In that case, the query of that user is rejected (i.e., if its k-trustee requirement is not met). Since only the query of that user fails, we believe its influence to other users is not big. Lastly, the cloaking process stops when all the users have been processed. Note that, to simplify, the restriction of the time $T_M^{u_i}$ defined by a user $u_i$ are not involved in this algorithm. We recommend that it be

**Algorithm 1:** Cloaking Algorithm Guaranteeing *k-Trustee*

```
Input: A road network $G(J, S)$, active users in a user set $U$
       traveling in $G$, a time instant $t$, and a privacy setting
       $p^{u_i}(k^{u_i}, e_l^{u_i}, f_l^{u_i}, e_g^{u_i}, f_g^{u_i}, R_M^{u_i}, T_M^{u_i})$ of each user $u_i \in U$
Output: An anonymized set $RS\langle u_i, R_t^{u_i}\rangle$
1  $CU \leftarrow \emptyset$;
2  $RS \leftarrow \emptyset$;
3  $es \leftarrow$ getExpansionScheme();
4  foreach $u_i \in U$ do
5      getTrustees($u_i$, $p^{u_i}$, $t$);
6  while $CU \neq U$ do
7      $u_i$ = pickAnUnprocessedUser($U$, $CU$);
8      $CR_t^{u_i} \leftarrow s_t^{u_i}$;
9      while $!PrivacyMet(CR_t^{u_i})$ do
10         $s_j \leftarrow$ getExpanded($CR_t^{u_i}$,$es$);
11         $CR_t^{u_i} \leftarrow CR_t^{u_i} + s_j$;
12     $R_t^{u_i} \leftarrow CR_t^{u_i}$;
13     foreach $u_j \in R_t^{u_i}$ do
14         $CU \leftarrow u_j$;
15         if $Size(R_t^{u_j}) > Size(R_M^{u_j})$ then
16             $R_t^{u_j} = unavaliable$;
17         else
18             $RS \leftarrow (u_j, R_t^{u_j})$;
```

handled by a user's mobile device.

Note that the guarantees of POI *l-Diversity* [4] and road segment *s-Diversity* [40] can be additionally ensured by the *k-Trustee* cloaking-based location privacy preserving mechanism. The *k-Trustee* cloaking process can first satisfy *l-Diversity* or *s-Diversity* requirement for every users in the region and then it satisfies the *k-Trustee* requirement. It can also first satisfy every user's *k-Trustee* requirement and then it guarantees the *l-Diversity* or *s-Diversity* requirement. We argue that the latter approach may be more appropriate when most of the users are strict to their trusted users by setting high values for *k-Trustee* requirements. It is because the guarantee of *k-Trustee* usually also ensures *l-Diversity* and *s-Diversity.* On the other hand, the former probably works more efficiently when there are fewer fake users in the road network and users are less strict in defining their trusted users.

## 4. Simulations

In this section, we first present the results of location injection attacks in the cloaking mechanism (called the general cloaking) guaranteeing only *k-Anonymity* and the one supporting both *k-Anonymity* and *s-Diversity* (referred as XStar [40]). We choose the general cloaking algorithm as the baseline approach and the XStar algorithm as the advanced approach. We want to demonstrate that the location injection attack is effective for both $k$-anonymity and $s$-diversity. We also want to compare the performance of a location injection attack when it is launched over a cloaking algorithm purely designed for $k$-anonymity and a cloaking algorithm designed for both $k$-anonymity and $s$-diversity. After evaluating the location injection attacks, we simulate the proposed *k-Trustee* cloaking algorithm and demonstrate its effectiveness against the location injection attacks.

### *4.1. Experiment Setup*

In our experiments, we use the GT Mobile simulator [37] to generate trajectories of 30,000 users moving in the DeKalb County in Atlanta regions of Georgia, which contains 37996 segments and 27647 junctions. We assume that each user is active during the travel in this road network and he has a location-based request from a specific location service provider every second. We also assume that AS, users, sensor agents at road segments can compute various intermediate results and communicate with each other instantly. The simulator runs for 10 minutes and each user has 600 location-based requests in total. Note that we assume that all of these 30,000 users are authentic users.

#### *4.1.1. Privacy Settings*

Given any authentic user $u$ in the road network, we first set $k^u$ for *k-Anonymity* and *k-Trustee* as a randomly chosen value from 2 to 10. We also set $r^u$ (used by XStar to support *s-Diversity*) as randomly chosen values between 2 and 5. The maximum size $R^u_M$ (the number of the road segments) of the cloaked region accepted by $u$ is a random value chosen from the set $\{20r^u, 30r^u, 40r^u, 50r^u\}$. We then set $e^u_l$ and $f^u_l$ as random values between 20 and 40, respectively. The global privacy requirements, $e^u_g$ and $f^u_g$ are both set as 5 for $u$. Note that, in the simulations, we assume that each user can get the cloaked results from AS immediately and we do not set the maximum waiting time ($T^u_M$) for $u$.

Regarding each fake user $fu$ created by the attacker, we choose the least privacy restrictions for him; *i.e.*, we set $k^{fu} = r^{fu} = 2$, $R^{fu}_M = 250$, $e^{fu}_l = f^{fu}_l = 40$, $e^{fu}_g = f^{fu}_g = 5$.

#### *4.1.2. Target Selection*

As shown in Section 2.4, the location injection attack has two types of targets: user targets and location targets. In the simulation, we randomly select 1,000 out of 30,000 authentic users as the user targets. Regarding location targets, we focus on the road segments which have at least one user during the simulation, *i.e.*, the average traffic of the road segment (the number of users visiting a road segment) is no less than 1. There are 9,033 such road segments in the dataset and we randomly select 1,000 of them as location targets. In addition, we choose 10 trajectories as the targets for the fixed-trajectory attack. These selected one consist of 10 connected road segments and there are total of 95 authentic users traveling on them.

#### *4.1.3. Fake User Creations*

We simulate fake users according to the proposed three attack models (refer to Section 2.4) as follows:

**Stalking Attack**. We assume that an attacker initially knows the initial location (a road segment) of the targeted user by injecting enough fake users into the road network (refer to Section 2.4.1). We then generate 6, 8 and 10 fake users, respectively, for a targeted user when the general cloaking is adopted. We also create 10, 15 and 20 fake users, respectively, for a targeted user when the *XStar* is applied. These fake users travel either in the same segment with the targeted user or few segment-based distant away from the targeted user as described in Section 2.4.1.

**Fixed-location Attack**. At a targeted road segment, we generate 2, 4 and 6 fake users, respectively, when the general cloaking is adopted. We also create 6, 8 and 10 fake users placed at a targeted road segment, respectively, when the *XStar* is applied. The location injection attack needs more fake users to compromise the *XStar* algorithm successfully. Unlike the general cloaking algorithm that only guarantee $k$-Anonymity, the *XStar* algorithm is designed for both $k$-Anonymity and $s$-Diversity. It is the $s$-Diversity that makes the *XStar* algorithm harder to be compromised. These fake users travel at the trajectory described in Section 2.4.2.

**Fixed-trajectory attack**. Given a targeted trajectory, we put 4, 6 and 8 fake users traveling at every road segment in the trajectory, respectively, in this attack. The goal of such an attack is to identify users who travel exactly on these trajectories.

Note that the times of the attacks conducted for each target is 600 in our simulation since each user has a total of 600 location-based requests during his travel in the road network (refer to 4.1).

### *4.1.4. Measurements*

In the simulations, we adopt the following measurements to evaluate the location injection attacks and the cloaking-based mechanism guaranteeing *k-Trustee*:

**In-distinguishability** $D_R$ : It indicates the in-distinguishability of a user (the number of trusted users) in a cloaked region. Each user has specified its required value ($k$) in the *k-Anonymity* and in the *k-Trustee* cloaking mechanisms. The location injection attack aims to compromise it by lowering its value. Note that fake users do not contribute to $D_R$ for an attacker since they are distinguishable for the attacker. A lower value of $D_R$ indicates the lower location privacy protection. Thus, the lower value of $D_R$ a location injection attack can achieve the more successful the attack is.

**Size of a Cloaked Region** $S_R$: It represents the in-distinguishability of a road segment in a cloaked region. In the *XStar*, each user specifies the privacy requirement for $S_R$. The location injection attacks may be able to lower its value. In addition, we use it to demonstrate the quality of service for the *k-trustee* cloaking-based mechanism.

**Cloaking Failure Rate** $F_R$: In the *XStar* and the *k-Trustee* cloaking mechanisms, a user usually needs to define the maximum size of the cloaked region, *i.e.*, the maximum number of road segments in a cloaked region. Given a user, we say the anonymity service is failed when the size of the cloaked region for a user is larger than the defined maximum size by the user. Then, $R_R$ generally indicates how practical the cloaking-based mechanism is.

**Attack Successful Rate** $A_R$: In an attack, given a user and a cloaked region, when the number of the trusted users in the cloaked region is smaller than a specified $k$ by the user, we say the attack is successful. The $A_R$ for a user indicates how successful the location injection attacks work for that user in general.

### *4.2. Attack Results of Location Injection Attacks*

In the subsection, we present the results of the location injection attacks on two road network-aware cloaking algorithms guaranteeing *k-Anonymity*: 1) the general cloaking algorithm with a random expansion; 2) *XStar* cloaking algorithm [40] which preserves users' location privacy with the additional guarantee of *s-Diversity*.

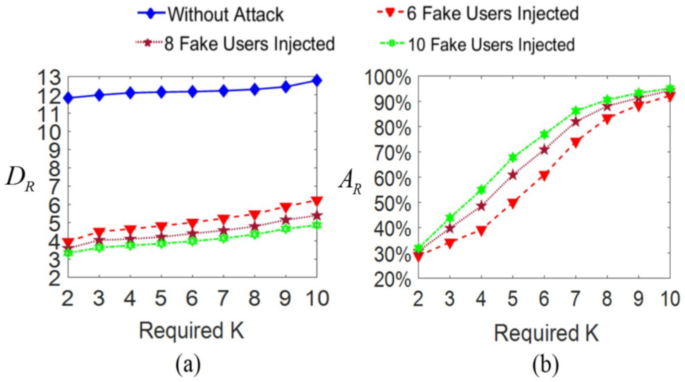


Figure 7: Stalking Attacks on the General Cloaking Algorithm

**Stalking Attacks on a General Cloaking Algorithm.** The attack results of the stalking attacks on the general cloaking algorithm are shown in Figure 7. It demonstrates the average $D_R$ (Figure 7.a) and the average $A_R$ (Figure 7.b) for targeted users with their diverse *k-Anonymity* requirements. From Figure 7.a, we can see that the stalking attacks can significantly downgrade the $D_R$ for targeted users. When the stalking attacks are not launched, the range of the $D_R$ for targeted users is between 12 and 13 with the guarantee of *k-Anonymity* for each user. However, when the attacks are launched, $D_R$ decreases to the range of 3 and 6. When the required $k$ in the *k-Anonymity* specified by targeted users is larger than 6, we even find that the average $D_R$ for these users is even lower than 6. Such a result suggests the successes of the attacks. Figure 7.b also confirms the successes of the attacks by showing that the average $A_R$ is more than 50% when the required $k$ of *k-Anonymity* is larger than 6. We also find that $A_R$ increases with the increase of the required $k$. Such a result reflects that the stalking attacks are more successful for users with the more restricted *k-Anonymity* requirements.

**Stalking Attacks on the *XStar* Algorithm.** Figure 8 shows the stalking attack results on the *XStar* cloaking algorithm. Figure 8.a demonstrates the average $A_R$ for targeted users with their various *k-Anonymity* requirements while Figure 8.b indicates the average $S_R$ with users' *s-Diversity* requirements. From Figure 8.a, we can see that $A_R$ has a significant increase from 5% to 80% with the increased value of the required $k$ in the *k-Anonymity.* Such a result shows that most attacks are successful. From Figure 8.b, we find that the attacks can dramatically decrease values of $S_R$ from 10 to 5 causing the targeted users easier to be distinguishable. However, since the required $S$ of *s-Diversity* defined by targeted users are between 2 and 5, the stalking attacks cannot actually compromise the guarantee of *s-Diversity.* Lastly, we also find that the number of fake users used in the attacks for the *XStar* cloaking algorithm may not be able to significantly promote the attack results from both graphs.

**Fixed-location Attacks.** The results of the fixed-location attacks on the general cloaking algorithm are shown in Figure 9. It shows that the average $D_R$ of users visiting every targeted road segment. We can see that $D_R$ significantly decreases with the increasing number of injected fake users. When 2 fake users are injected at a targeted segment, it seems that the *k-Anonymity* requirements for

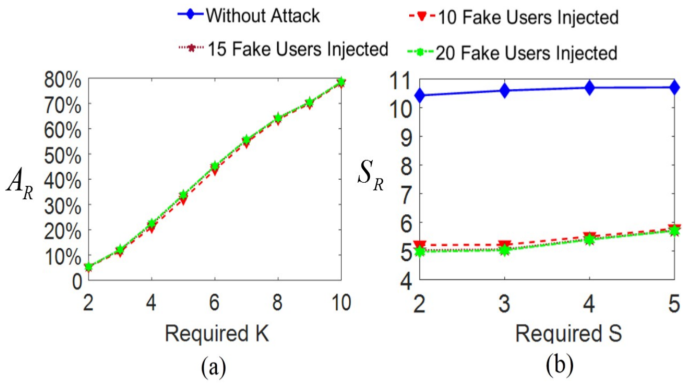


Figure 8: Stalking Attacks on the XStar Algorithm

users visiting the targeted segment are still satisfied in most attack instances. However, when 4 or 6 fake users are placed at a targeted segment, $D_R$ deceases significantly than the required value and users' *k-Anonymity* requirements are compromised. In our simulation, we also find that $A_R$ is between 20% and 40% when 2 fake users are placed. When 4 or 6 fake users are injected, the range of $A_R$ has a remarkable increase and it is between 60% and 90%. These results also confirm the successes of the fixed-location attacks on the general cloaking algorithm. In addition, we simulated the fixed-location attacks on the *XStar* cloaking algorithm. However, such attacks are less successful and the average $A_R$ is below 20%. We believe that the enforcement of the *s-Diversity* can mitigate the fixed-location attacks to some extent.

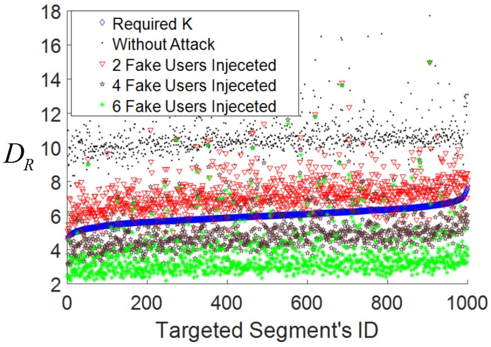


Figure 9: Fixed-location Attacks on the General Cloaking Algorithm

**Fixed-trajectory attacks.** We also performed the fixed-trajectory attacks for the chosen trajectories in order to identify users who follow these trajectories. The results of the attacks on a general cloaking algorithm are shown in Figure 10. Among the 95 targets in the fixed-trajectory attacks, the percentage of compromised users rises from 0.52 to 0.96 when the number of placed fake users are increased from 4 to 8. These numbers reflect the successes of the fixed-trajectory attacks. We also simulated the fixed-trajectory attacks under the *XStar* clocking algorithm. However, the attacks are not successful and we cannot identify the trajectory of any user in terms of the requirements of *s-Diversity* defined by users. To have a successful fixed-trajectory attack, all the constructed cloaked regions from AS for a user should have only one road segment in order to determine the user's trajectory. The cloaked region from the *XStar* includes more than one road segment and hence the attack cannot be successful.

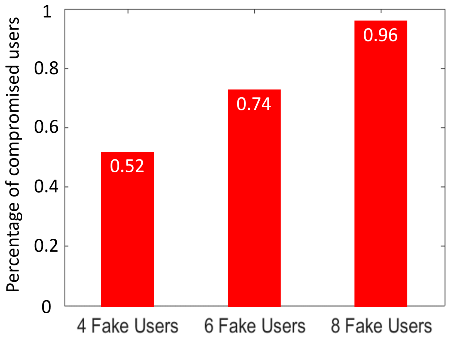


Figure 10: Results of fixed-trajectory attacks

### *4.3. Location Injection Attacks on k-Trustee Cloaking Algorithm*

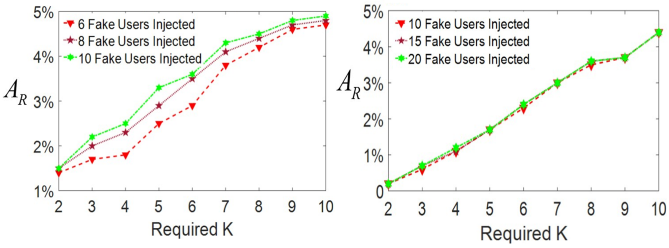
In this subsection, we utilize the same fake users for the same targeted users and locations as those used in the Section 4.2. We simulate various location injection attacks on the *k-Trustee* cloaking mechanism. Note that, in these simulations, when users additionally specify requirements for the diversity of the road segments, we first guarantee their *k-Trustee* requirements and then meet those for the diversity of road segments.

#### *4.3.1. General Attack Results*

We first perform location injections for targets on the *k-Trustee* cloaking mechanism adopting the coarse-grained trust functions and the random expansion scheme.

Figure 11 shows the results of the stalking attacks on the *k-Trustee* cloaking mechanism. Figure 11.a indicates the average $A_R$ for targeted users who has different *k-Anonymity* requirements and do not specify the diversity of road segments. Figure 11.b demonstrates the average $A_R$ for targeted users who do specify the diversity of road segments. From this figure, we can see that less than 5% of the stalking attack instances on the *k-Trustee* cloaking mechanism are successful. Compared to those successful rates shown in Figures 7 and 8, we can conclude that the *k-Trustee* cloaking mechanism can significantly defend against the stalking attacks.

In addition, our simulation results for the fixed-location attacks on the *k-Trustee* cloaking mechanism demonstrate that less than 4% of the attack instances on average are successful for targeted road segments when users do not specify the diversity of road segments. When users do specify this requirement, our results show that the average $A_R$ is less than 1.5%. Compared to the corresponding results of the same attacks on the general cloaking algorithm in Section 4.2, we can say that the fixed-location



(a) Users not specifying the diversity of road segments

(b) Users specifying the diversity of road segments

Figure 11: Results of the Attacks on the *k-Trustee* Cloaking Mechanism

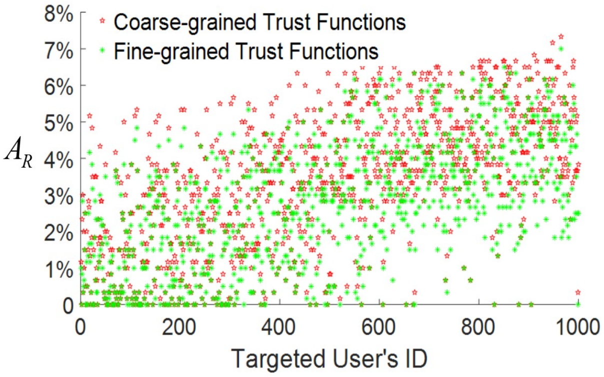


Figure 12: Coarse-grained Trust Functions *vs.* Fine-grained Trust Functions

attacks become significantly less successful when the *k-Trustee* cloaking mechanism is applied. Furthermore, we performed the fixed-trajectory attacks on the *k-Trustee* cloaking mechanism. However, we cannot identify any user's trajectory in this case.

Based on the above results, we can conclude that the *k-Trustee* cloaking mechanism is indeed effective to mitigate the location injection attacks.

### *4.3.2. k-Trustee Cloaking Mechanism with Different Trust Functions*

We next compare the effectiveness of the *k-Trustee* cloaking mechanisms using different trust functions (coarse-grained trust and fine-grained trust functions) as discussed in Section 3.2. We focus on the stalking attacks using 8 fake users for each targeted users. We also assume that a random expansion is adopted in these *k-Trustee* cloaking mechanisms and users do not specify the diversity of road segments. We then perform the stalking attacks for the targeted users using different trust functions in the *k-Trustee* cloaking mechanism and the results are shown in Figure 12. It shows the average $A_R$ for each targeted users. We can see that both of the coarse-grained and fine-grained trust functions adopted by the *k-Trustee* cloaking mechanisms are effective to mitigate the location injection attacks. As expected, the fine-grained trust functions can achieve a better resilience by approximately decreasing 1% of $A_R$ by average.

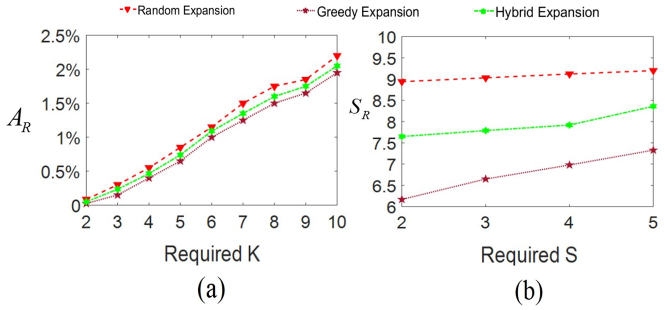


Figure 13: Results of the Attacks using Different Expansion Schemes in the *k-Trustee* Cloaking Mechanism

### *4.3.3. k-Trustee Cloaking Mechanism with Different Expansion Schemes*

In Section 3.3.2, we discussed three different expansion schemes for the *k-Trustee* cloaking mechanism and we compare them in this subsection. We first focus on the stalking attacks using 10 fake users for each targeted user who also specifies the diversity of road segments. We then adopt the coarse-grained trust functions for the *k-Trustee* cloaking mechanisms using the random expansion, the greedy expansion and the hybrid expansion. Our results are shown in Figure 13. Figure 13.a demonstrates the average $A_R$ for targeted users with different *k-Anonymity* requirements. We can see that attacks on the the *k-Trustee* cloaking mechanism using the greedy expansion can achieve the lowest $A_R$ while those on the *k-Trustee* cloaking mechanism adopting the random expansion have the highest $A_R$. Figure 13.b indicates the average size of cloaked regions, $S_R$, for targeted users with different requirements for the diversity of road segments. We can find that the random expansion induces AS to construct the largest cloaked regions while the greedy expansion induces AS to construct the smallest cloaked regions. Based on these results, we can say that the greedy expansion has the best resilience against location injection attacks and it can achieve the best quality of the location-based services.

### *4.4. Discussion*

In our experiments, we noticed that the location injection attacks are not successful for some user and location targets. We analyzed these targets and found out that there are always enough number of users traveling with the targeted users and/or traveling on the targeted segments. As a result, the location injection attacks are not successful. In addition, the *k-Trustee* cloaking mechanism using fine-grained functions may have the performance issue due to the complexity of algorithm. Such a mechanism needs to maintain two relatively larger matrices: user to user matrix which dynamically records the trust between each pair of users based on their distance, and location to user matrix that dynamically calculates the trust between each pair of road segment and user based on their distance. We presented a framework with the additional sensor agent at every road segment in Section 3.3.1, which should be able to improve the performance of the proposed *k-Trustee* mechanism.

## 5. Related Work

Location privacy has been an active area of research for decades. To protect users' location privacy during usage of location based services (LBS), various location privacy protection mechanisms have been proposed. Based on their core ideas, these mechanisms can be broadly categorized into approaches that use dummies [29, 32], space transformation [18, 28], mix-zone [6, 36], encryption [31], spatial cloaking [5, 10, 17, 20, 21, 27, 30, 34, 40, 43] and differential privacy [3, 22, 42]. The basic ideas behind these techniques are briefly discussed as follows. The dummy-based approaches replace real user locations with fake locations that are related to the real ones. The schemes based on spatial transformation transform data to another space to encode relationship between data and queries. The mix-zone solutions change pseudonyms of users who enter the zones so that adversaries are unable to link leaving users with entering users. The encryption-based schemes use cryptographic techniques to protect privacy of location data. For instance, in [31], Li *et al.* applied CP-ABE [7] to extend binary access to location data to a fine-grained access control model. The spatial cloaking mechanisms, as the most widely studied category, usually generate cloaked regions that satisfy privacy requirements such as $k$-anonymity [20] for users and send such cloaked regions to LBS providers. More recent work have introduced the newer privacy paradigm of differential privacy [16], to location privacy protection [3, 22, 42]. By carefully applying differential privacy protection mechanisms (e.g. Laplace Mechanism [16], Exponential Mechanism [33]) to the location data, the personal location information in the disclosed statistical output can be protected. Among these techniques, we have focused on studying the spatial cloaking technique in this work because it is the one that has been widely studied with respect to various settings (e.g., centralized [17, 34], P2P [13, 14]) as well as various problem statements (e.g., snapshot queries [17], trajectories [12]). While differential privacy provides a more formal and rigorous privacy guarantee against background knowledge attacks, it can result in a higher perturbation and may provide a lower data utility compared to spatial cloaking techniques. Thus, in cases where there is a lack of background knowledge and when the risks of such attacks are minimal, the spatial cloaking techniques are likely to provide a higher data utility compared to differential privacy. A unified framework for location privacy that offers a systematic view by formalizing the problem, adversaries, mechanisms and metrics can be found in [39].

The notion of spatial cloaking was first introduced by Beresford and Stajano [5]. From then on, many centralized approaches have been proposed, which essentially leverage a centralized anonymization server to deploy the spatial cloaking algorithms. Among these approaches, Gruteser and Grunwald [20] presented the *Interval Cloak* that guarantees *k-Anonymity* in the cloaked region to preserve users' location privacy from LBS providers. Gedik and Liu [17] introduced the *CliqueCloak* where users' personalized privacy requirements for *k-Anonymity* are satisfied. Mokbel *et al.* [34] designed *Casper* that extended the *Interval Cloak* to the grid network with the privacy-aware query processor. Hoh *et al.* [21] developed a time-to-confusion criterion as the duration over which an attacker could track a target. Based on it, they designed an uncertainty-aware path cloaking mechanism that guarantee *k-Anonymity* for all users and hide users' trajectories. Kalnis *et al.* [27] improved the previous cloaking algorithms by introducing the *Hilbert Cloak*. The *Hilbert Cloak* satisfies reciprocity that is sufficient for users to achieve the spatial *k-Anonymity* for their location requests. Cui *et al.* [15] extended the *Hilbert Cloak* by considering average query density to make anonymity set satisfy both reciprocity and uniformity. Zheng *et al.* [45] proposed an approach that selects a sub-area from the clocked region that may or may not include the real user location to prevent side information attacks launched by adversaries.

However, centralized approaches usually suffer from a single point of trust, which motivates the research of decentralized solutions that do not need the anonymization server. As the representative solution, Chow *et al.* [13] proposed a peer-to-peer (P2P) spatial cloaking algorithm that leverages single-hop communication and/or multihop routing among peers to generate cloaked region without help from a centralized anonymization server. The algorithm offers two modes. The candidate searching step is triggered by queries in the *demand* mode, whereas it is periodically executed in the *proactive* mode. Later, Chow *et al.* [14] improved their scheme with information sharing scheme, historical location scheme and cloaked region adjustment scheme. After that, Che *et al.* [8] proposed the *dual-active* mode that allows peers both actively collect location data and actively disseminate collected data to others, which offers better performance than the previous two modes. However, the above P2P approaches are not reliable when there are malicious peers in the network. To secure the P2P scheme, Jin *et al.* [23, 24] introduced the pseudonymous authentication technique to provide message authentication and integrity for peer communication, thus significantly suppressing the impact of malicious peers.

Recent work has considered the location cloaking problem under a constrained road network model [40, 30, 10, 43]. Wang and Liu [40] implemented *XStar* which supports the *k-Anonymity* and the road segment diversity in a road network. Li and Palanisamy [30] further made the $k$-anonymity reversible. However, all of these algorithms guarantee *k-Anonymity* but they are vulnerable to the proposed location injection attacks as shown in Section 2.3.

The fake users/accounts/identities have become a well-known security and privacy issue [38]. This problem can also pose a threat to data aggregation systems, voting systems, peer-to-peer systems, social networks and misbehavior detection mechanisms. For example, in the peer-to-peer systems, such a problem can lead to the *Sybil attacks* where an attacker forges multiple identities to compromise the network to arbitrarily subvert content storage and acquisition [38]. In social networks, an attacker can create fake accounts to impersonate victims, deceive the victim's friends and destroy the victims' reputations [26]. In the literature, the defense approaches against these attacks are usually based on trust among users, position verification, game theory and access control mechanisms. For instance, Yu *et al.* proposed a *Sysbil* defense approach based on the trust in the social networks [44]. Chen *et al.* [9] proposed a generalized attack-detection model using the spatial correlation of RSS inherited from wireless nodes to detect Sybil nodes. In this paper, we identify that fake users can also be utilized in the cloaking-based privacy preserving mechanisms to compromise the guarantee *k-Anonymity* via the proposed location injection attacks. We then design the *k-Trustee* cloaking-based mechanisms to mitigate such attacks. To the best of our knowledge, our proposed approach is the first work to address this kind of attacks in the cloaking-based privacy preserving mechanisms.

## 6. Conclusion

In this paper, we identified the vulnerability of the existing cloaking-based location privacy preserving mechanisms and showed the location injection attacks against them. We proposed various attack models and demonstrated the effectiveness of these attacks through simulations. We then proposed the cloaking-based mechanisms that guarantee the notion of *k-Trustee* by employing different trust functions and expansion schemes to mitigate the location injection attacks. Through simulations, we demonstrated that the *k-Trustee* cloaking-based privacy preserving mechanisms are effective against these attacks. As future work, we plan to study how to achieve *k-Trustee* in a peer-to-peer (P2P) environment that has no centralized anonymization server. Since P2P spatial cloaking algorithms usually leverage P2P communication to build the cloaked region, the generation of fake users become harder. However, similar attacks can still be launched by malicious users. For example, a fixed-location attack can be launched by malicious users that stay in a particular location, e.g., a hospital. Such an adversary can communicate with nearby peers to form cloaked regions and learn about the peers who are visiting the hospital. Because of the lack of a global view, computing global trustees in the P2P environment becomes a challenging problem. A promising solution to the lack of a global view is to leverage the blockchain [35] technique to build global trust. We believe it can be adopted as a digital ledger to record the *e-stalker* and *f-stationary* to offer a trusted global view in the P2P environment. A blockchain insures credibility so that all the users are guaranteed that they all see the same *e-stalker* and *f-stationary* when they participate in the location anonymization process. These values, once being submitted to the blockchain, become nearly tamper-proof unless someone controls a majority of computation power of the distributed network [1]. One potential way to implement this process is to develop a decentralized application over the Ethereum smart contract platform [41], which can collect *e-stalker* and *f-stationary* from mobile users, compute the global *e-stalker* and *f-stationary* values and show these global values to all the mobile users through Ethereum mobile browsers (e.g., Toshi [2]). Another future direction is to enable our system to distinguish intentional stalking behavior from unintentional stalking. A group of people who travel together within a period of time may label each other as *e-stalkers*. As a result, each of them may be globally labeled as *e-stalker* by many, and thus, it becomes hard to add them into a cloaked region that require members with lower count of being an e-stalker. A potential solution requires each user to maintain a list of trusted user IDs, such as a friend list or family group in many kinds of applications. Then, after receiving the cloaked regions from AS, this list can be used as a filter so that the *e-stalker* values are only computed for the unknown users. Here, it is important that the metadata used should not create additional privacy risks. In order to control and mitigate such potential risks, one approach is to avoid the creation of new metadata each time for the location anonymization process and instead we can leverage existing user relationship information such as a friend list on social networks as the source of metadata. Such an approach significantly reduces the amount of newly generated metadata and mitigates any potential risks associated with the use of metadata.

[1] Ethernodes: The ethereum node explorer. `https://www.ethernodes.org/network/1`.

[2] Toshi. `https://www.toshi.org/`.

[3] 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. ACM, 2013.

[4] Bhuvan Bamba, Ling Liu, Peter Pesti, and Ting Wang. Supporting anonymous location queries in mobile environments with privacygrid. In *Proceedings of the 17th international conference on World Wide Web*, pages 237–246. ACM, 2008.

[5] Alastair R Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive computing*, 2(1):46–55, 2003.

[6] Alastair R Beresford and Frank Stajano. Mix zones: User privacy in location-aware services. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 127–131. IEEE, 2004.

[7] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 321–334. IEEE, 2007.

[8] Yanzhe Che, Qiang Yang, and Xiaoyan Hong. A dual-active spatial cloaking algorithm for location privacy preserving in mobile peer-to-peer networks. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 2098–2102. IEEE, 2012.

[9] Yingying Chen, Jie Yang, Wade Trappe, and Richard P Martin. Detecting and localizing identity-based attacks in wireless and sensor networks. *IEEE Transactions on Vehicular Technology*, 59(5):2418–2434, 2010.

[10] Hyung-Ju Cho, Se Jin Kwon, Rize Jin, and Tae-Sun Chung. A privacy-aware monitoring algorithm for moving k-nearest

neighbor queries in road networks. *Distributed and Parallel Databases*, 33(3):319–352, 2015.

[11] Chi-Yin Chow and Mohamed F Mokbel. Enabling private continuous queries for revealed user locations. In *International Symposium on Spatial and Temporal Databases*, pages 258–275. Springer, 2007.

[12] Chi-Yin Chow and Mohamed F Mokbel. Trajectory privacy in location-based services and data publication. *ACM Sigkdd Explorations Newsletter*, 13(1):19–29, 2011.

[13] Chi-Yin Chow, Mohamed F Mokbel, and Xuan Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 171–178. ACM, 2006.

[14] Chi-Yin Chow, Mohamed F Mokbel, and Xuan Liu. Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. *GeoInformatica*, 15(2):351–380, 2011.

[15] Ningning Cui, Xiaochun Yang, and Bin Wang. A novel spatial cloaking scheme using hierarchical hilbert curve for location-based services. In *International Conference on Web-Age Information Management*, pages 15–27. Springer, 2016.

[16] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[17] Bugra Gedik and Ling Liu. Location privacy in mobile systems: A personalized anonymization model. In *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*, pages 620–629. IEEE, 2005.

[18] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. Private queries in location based services: anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 121–132. ACM, 2008.

[19] Gabriel Ghinita, Panos Kalnis, and Spiros Skiadopoulos. Prive: anonymous location-based queries in distributed mobile systems. In *Proceedings of the 16th international conference on World Wide Web*, pages 371–380. ACM, 2007.

[20] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM, 2003.

[21] Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 161–171. ACM, 2007.

[22] Jingyu Hua, Wei Tong, Fengyuan Xu, and Sheng Zhong. A geo-indistinguishable location perturbation mechanism for location-based services supporting frequent queries. *IEEE Transactions on Information Forensics and Security*, 2017.

[23] Hongyu Jin and Panos Papadimitratos. Resilient collaborative privacy for location-based services. In *Secure IT Systems*, pages 47–63. Springer, 2015.

[24] Hongyu Jin and Panos Papadimitratos. Resilient privacy protection for location-based services through decentralization. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 253–258. ACM, 2017.

[25] Lei Jin, Balaji Palanisamy, and James BD Joshi. Poster: compromising cloaking-based location privacy preserving mechanisms with location injection attacks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 1439–1441. ACM, 2014.

[26] Lei Jin, Hassan Takabi, and James BD Joshi. Towards active detection of identity clone attacks on online social networks. In *Proceedings of the first ACM conference on Data and application security and privacy*, pages 27–38. ACM, 2011.

[27] Panos Kalnis, Gabriel Ghinita, Kyriakos Mouratidis, and Dimitris Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE transactions on knowledge and data engineering*, 19(12):1719–1733, 2007.

[28] Ali Khoshgozaran and Cyrus Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In *International Symposium on Spatial and Temporal Databases*, pages 239–257. Springer, 2007.

[29] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. Protection of location privacy using dummies for location-based services. In *Data Engineering Workshops, 2005. 21st International Conference on*, pages 1248–1248. IEEE, 2005.

[30] Chao Li and Balaji Palanisamy. Reversecloak: Protecting multilevel location privacy over road networks. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 673–682. ACM, 2015.

[31] Xiang-Yang Li and Taeho Jung. Search me if you can: privacy-preserving location query service. In *INFOCOM, 2013 Proceedings IEEE*, pages 2760–2768. IEEE, 2013.

[32] Hai Liu, Xinghua Li, Hui Li, Jianfeng Ma, and Xindi Ma. Spatiotemporal correlation-aware dummy-based privacy protection scheme for location-based services. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2017.

[33] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103. IEEE, 2007.

[34] Mohamed F Mokbel, Chi-Yin Chow, and Walid G Aref. The new casper: Query processing for location services without compromising privacy. In *Proceedings of the 32nd international conference on Very large data bases*, pages 763–774. VLDB Endowment, 2006.

[35] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[36] Balaji Palanisamy and Ling Liu. Attack-resilient mix-zones over road networks: architecture and algorithms. *IEEE Transactions on Mobile Computing*, 14(3):495–508, 2015.

[37] P Pesti, B Bamba, M Doo, L Liu, B Palanisamy, and M Weber. Gtmobisim: A mobile trace generator for road networks. *College of Computing, Georgia Inst. of Tech*, 2009.

[38] Hosam Rowaihy, William Enck, Patrick McDaniel, and Tom La Porta. Limiting sybil attacks in structured p2p networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2596–2600. IEEE, 2007.

[39] Reza Shokri, Julien Freudiger, and Jean-Pierre Hubaux. A unified framework for location privacy. Technical report, 2010.

[40] Ting Wang and Ling Liu. Privacy-aware mobile services over road networks. *Proceedings of the VLDB Endowment*, 2(1):1042–1053, 2009.

[41] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151, 2014.

[42] Yonghui Xiao, Li Xiong, Si Zhang, and Yang Cao. Loclok: location cloaking with differential privacy via hidden markov model. *Proceedings of the VLDB Endowment*, 10(12):1901–1904, 2017.

[43] Bidi Ying and Dimitrios Makrakis. Protecting location privacy with clustering anonymization in vehicular networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pages 305–310. IEEE, 2014.

[44] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 267–278. ACM, 2006.

[45] Jiangyu Zheng, Xiaobin Tan, Cliff Zou, Yukun Niu, and Jin Zhu. A cloaking-based approach to protect location privacy in location-based services. In *Control Conference (CCC), 2014 33rd Chinese*, pages 5459–5464. IEEE, 2014.