LISA: Location Information ScrAmbler for Privacy Protection on Smartphones

Zhigang Chen
Twitter Inc.
zhigangc@umich.edu

Xin Hu
IBM Research
huxin@us.ibm.com

Xiaoen Ju and Kang G. Shin
The University of Michigan
{jux,kgshin}@eecs.umich.edu

***Abstract*— As use of location-based services (LBSs) is becoming increasingly prevalent, mobile users are more and more enticed to reveal their locations, which may be exploited by attackers to infer the points of interest (POIs) the users visit and then their privacy information. We propose a novel approach to the protection of a user's location privacy based on *unobservability*, preventing the attackers from relating any particular POI to the user's current location. We design, implement, and evaluate a privacy-protection system, called the *Location Information ScrAmbler* (LISA) which protects the user's location privacy by adjusting the location noise and hence, the uncertainty of associating his location with any POI, while conserving resources (especially battery energy) on mobile devices. By protecting location privacy *locally* on each mobile user's device, LISA eliminates the reliance on the trusted third-party servers required by most existing approaches. Therefore, it not only avoids the vulnerability of a single point of failure, but also facilitates the deployment of LBSs. Our evaluation of LISA using real-world users' traces demonstrates its efficacy and efficiency.**

I. Introduction

As GPS-integrated smartphones are becoming increasingly popular, location-based services (LBSs) have attracted significant attention from mass media [35], [38], financial investors [27], companies [3], [6], [29], and a rapidly-growing number of application developers and customers. LBS is an information service that exploits the knowledge about users' geographic locations to deliver personalized information tailored to their needs. Typical examples include turn-by-turn navigation to an addressed location, finding nearby business or service, receiving alerts such as warning of traffic jams in certain areas, and locating friends or events in the user's vicinity. According to a report by Computer Science and Telecommunications Board [7], LBSs are expected to be seamlessly and ubiquitously integrated into future computing environments and users' daily lives and businesses.

However, with the rapidly-growing deployment and use of LBSs, mobile (smartphone) users are enticed to provide their location. In particular, some LBSs, such as Dokiru and Geolife [26], and Alohar's Placeme [2], rely on real-time and continuous updates of location information, thus significantly risking mobile users' location privacy. For example, a user may use a location-based social network service to locate friends [11] and interesting events within his vicinity by running an application on his smartphone—as the smartphone user moves, the application periodically updates his location to the service and fetches a list of friends and events inside the moving geographic boundaries. Suppose the user is heading toward a clinic or a church, his phone, with or without his knowledge, still interacts with the service. Attackers or unauthorized entities, once obtained the trace, can infer his visits to the church/clinic (i.e., *Points of Interest* (POIs)), and thus his personal information, such as health condition or religious affiliations.

The privacy threat imposed by the use of LBSs has been attracting significant attention from both academia, industry [10], and the administration [1]. Most approaches proposed in academia to date for the protection of mobile users' location privacy are based on location perturbation and obfuscation, employing well-established generic privacy metrics such as k-anonymity and relying on a third-party trusted anonymization server for the proper functionality of the proposed protection mechanisms. The reliance on a third-party anonymization server can result in the susceptibility to a single point of failure and also raise similar privacy concerns for the anonymization server. If a mobile user is concerned about his location privacy and unwilling to reveal his location to LBS servers, they will also likely be reluctant to entrust his location information to a third-party server.

In this paper, we propose a novel approach called the *Location Information ScrAmbler* (LISA), to protection of mobile users' location privacy without requiring a third-party anonymization server. The key idea behind LISA is to make attackers less certain about which POI a mobile user visits next from his current location, and therefore, weaken their capability of inferring the user's privacy information. Our rationale is that a user's location privacy is often associated with POIs (or "identifying locations" [26]) and attacks on the location privacy often aim to infer *sensitive* POIs the user visits. For example, a user's home location, religion, and health condition may be revealed by his visits to residences, churches, and clinics. Consequently, one way to protect a user's location privacy is to prevent attackers from associating his current locations with sensitive POIs, i.e. the attacker's inability to distinguish which POI a mobile user visits.

To measure users' location privacy, LISA uses *unobservability*, which is calculated as the entropy (or uncertainty) that a location is related to a set of POIs. A location is said to be *m-unobservable* if and only if the information leakage is equivalent to that at least m POIs can be equally likely related to the location, i.e., the entropy is no less than $\log_2 m$. In order to protect a user's location privacy, LISA intentionally introduces a certain level of measurement noise into the locations provided to a LBS, such that they are *m-unobservable*. By using a simple yet general object-tracking model based on an extended Kalman filter [28], LISA adjusts the level of location

noise to meet the *m-unobservability* requirement. Since LISA performs such noise-level tuning *locally* on individual mobile handsets, it eliminates the reliance on trustworthy third-party servers. In LISA, mobile users need to trust only their handheld devices and can set up personalized privacy requirements. Its less demanding trust requirements and improved configuration flexibility significantly reduce the complexity of the design and deployment of LBSs, thus making LISA more attractive to privacy-concerned users.

The main contributions of LISA are four-fold. It

- introduces a new orthogonal dimension of uncertainty and can be combined with existing approaches to provide stronger location privacy protection;
- saves resource consumption by searching for the minimum level of noise to be introduced into the location data to satisfy the m-unobservability requirement, and by intelligently planning where LBS requests are sent;
- prevents leakage of mobile users' privacy information as a result of compromised third-party servers, and limits the impact of privacy attacks to individual smartphones;
- lowers the trust requirements from mobile users, thus simplifying the implementation and deployment of LBSs.

The paper is organized as follows. Section II reviews the previous work. Section III and IV describe our threat model and the privacy model. Section V presents the details of LISA, and Section VI describes two optimizations for performance improvement. Section VII discusses our evaluation and Section VIII concludes the paper.

II. Related Work

In this section we discuss prior work on location privacy in terms of threats, metrics, and protection mechanisms.

A. Location-Privacy Threats

There are two major types of LBS-related privacy: query privacy and location privacy. Query privacy refers to users' private information related to LBS query attributes. Typical threats related to query privacy include (1) inferring a user's identity (e.g., [12]) and (2) inferring a user's interests and habits from query contents (e.g., [33]). Location privacy refers to users' private information directly related to their locations, as well as other private information that can be inferred from the location information. Example threats (e.g., [19], [25]) in this category are (1) locating a user and (2) inferring a user's interests and habits based on his/her locations. Our work aims at mitigating location privacy threats, and below we provide a detailed analysis of this attack category.

Prior work targeting at location privacy threats mainly considered two types of attack: location disclosure and movement tracking. For the former, Hoh *et al*. [19] and Krumm [23] showed that a driver's home location can be inferred from GPS data collected on his vehicle even if the location data were pseudonymized. Moreover, Matsuo [25] exploited a user's indoor location data to infer a variety of his personal information, such as work role, smoker, coffee drinker, and age. For the latter, Gruteser and Hoh [15], [17] showed that coherent, individuals' traces can be reassembled from completely anonymized GPS data from three to five users by applying Multiple Hypotheses Tracking (MHT) algorithm.

B. Location-Privacy Metrics

There has not yet been any standard for the quantification of location privacy. Most of the location-privacy metrics to date are uncertainty-based. That is, a user's location privacy is better protected if attackers are made unable, or less able to differentiate the user from others within an anonymity set [14], linking two pseudonyms of the user outside a mix zone [4] or in a wireless LAN [21], or distinguishing paths along which a user may travel [26]. Therefore, the degree of location privacy is determined by the size of the anonymity set e.g. k-anonymity, the number of users in the mix zone, or the probability that the user is at a certain location. A good quantitative metric of uncertainty is the location entropy which has widely been adopted [18], [21], [26]. Aside from the uncertainty-based metrics, Hoh *et al*. [17] used the expected error between the attackers' estimation and the true location of a user to measure the user's privacy. A similar but more general use of the adversary's estimation error for quantifying location privacy is proposed in [37].

C. Location-Privacy Protection Mechanism

Most approaches to protecting location privacy employ perturbation and obfuscation. But other types of defense, such as policy-based schemes (e.g., [39]) and private information retrieval (PIR) based approaches (e.g., [13], [30]), have also been investigated.

Most location perturbation and obfuscation schemes assume the existence of a trusted anonymization server, where users' LBS queries are collected, anonymized and then transferred to LBS servers. Gruteser and Grunwald [14] designed an adaptive interval cloaking algorithm which replaces users' accurate locations with spatio-temporal cloaking boxes containing at least k_{min} users. Gedik and Liu [12] proposed CliqueCloak which achieves k-anonymity by enlarging the exposed spatial area (spatial cloaking) and delaying the query messages (temporal cloaking) until at least k different queries have been sent from the specific area. Further extensions of k-anonymity based approach include (1) taking both historical and current locations into consideration [40] and (2) employing more realistic adversary models (e.g., "policy-awareness" of adversaries) [9].

In addition, approaches based on *unlinkability* aim at unlinking the two pseudonyms of a user, preventing the attackers from accumulating enough history of the same user to infer his personal information. To achieve unlinkability, Beresford and Stajano [4] proposed the concept of mix zone in which a number of users simultaneously change to new, unused pseudo-names so that an external viewer cannot link people going into the zone with those coming out of it. Jiang and Wang [21] unlink different pseudonyms of the same user with silent periods between different pseudonyms, which are planned using well-known user mobility patterns.

For path confusion, Hoh and Gruteser [17] developed a data-perturbation technique that modifies the location data reported from users in close proximity such that the users' paths intersect with one another. Meyerowitz and Choudhury [26] proposed *CacheCloack*, a system that uses a mobility model to predict a user's future path and performs prospective path confusion without degrading accuracy. CacheCloak uses a centralized anonymization server to cache the LBS query results for the predicted paths. In case of a cache miss, a new predicted path is generated such that both ends of the new path intersect with existing paths. This way, the adversaries will be prevented from determining the user's exact path.

One significant drawback of most existing work is the requirement for a centralized trusted anonymization server, namely, the susceptibility to single-point failures and the trustworthiness of the server. Researchers thus explore protection schemes that are applicable to users' mobile devices. For example, CAP [32] used a quadtree to maintain road-density information and conducted the Various-grid-length Hilbert Curve (VHC) mapping and perturbation to achieve k-anonymity. In [20], k-anonymous cloaking boxes are generated by employing the nearby mobile devices to building the proximity information among the users via the received signal strength or the time difference of arrivals. However, this approach may become ineffective if the query sender cannot detect a sufficient number of users in the vicinity. Kido *et al*. [22] investigated ways to hide real user movements with dummies. Users' location Despite this success, a malicious LBS server may be able to differentiate the real user from those dummies after long-term movement tracking.

In this paper, we propose a very different but effective privacy-protection approach, called LISA, using the concept of *m-unobservability* [31]. LISA scrambles users' locations such that the estimated location is *m-unobservable*, i.e., the uncertainty of relating users to POIs is equivalent to the entropy that the user can be equally related to at least m POIs. Our design is based on the empirical fact that, in many cases, users' privacy is leaked by their strong association with a particular POI (e.g., home or hospital), if attackers cannot reliably associate any particular POI to a user's location, their ability to infer the user's privacy information is significantly weakened. LISA introduces a new dimension of uncertainty, and can also be combined with existing approaches enhance privacy protection. Moreover, LISA does not require any trusted third-party server for location-privacy protection, thus avoiding a single point of failure. Additionally, it greatly lowers the trust requirements from mobile users, significantly simplifying the implementation and the deployment of LBSs.

III. System and Threat Models

A typical LBS system consists of smartphones, wireless networks, and LBS servers, as shown in Fig. 1. A smartphone user moves from one POI to another, performing daily activities such as going to work (from home to office) and seeing a doctor (from office or home to a hospital). At the same time, the smartphone user may access a LBS by sending requests to a certain LBS provider via wireless networks. The user's requests include information about his current location, obtained from the GPS integrated in his phone or by triangulation of nearby radio tower locations. The provider then returns the queried information to the user. The communication between the mobile device and LBS servers can be done through secure connections to prevent unauthorized eavesdropping.

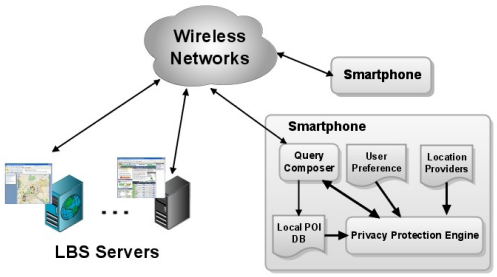


Fig. 1: A typical LBS system architecture

Here we adopt a *generic* threat model in which adversaries may compromise any third-party servers (e.g., LBS servers or anonymization servers) and gain unauthorized access to *any* information in users' LBS service requests, such as their identity (names and IP addresses used in the LBS service), current locations, and past trajectories. Adversaries can thus keep track of the POIs a user has visited or intends to visit, and infer a wide range of privacy information of the user, such as home or private offices [26]. As a result, we assume that a user trusts only his own handset (See [5] for an efficient way of self-securing handsets.) and therefore, *cannot* rely on third-party servers (such as the location server [14] or the anonymization server [12]) to protect his location privacy. This model removes the dependency on trusted third-party servers, and is thus more realistic and attractive.

As in previous work [12], [14], [18], we also assume that adversaries only make use of the location information in the compromised servers to breach the users' privacy and do not manipulate the responses from LBS servers to the users. This is because attackers' main purpose is to learn users' location privacy information. Although modifying responses may allow attackers to mislead mobile users to a wrong place, it provides little help for attackers who try to learn the users' privacy information. In addition, bogus responses also increase the risk of attackers to be caught, as users can easily verify the validity of a server's responses with their expectation and observation.

IV. Privacy Model and Metrics

LISA adopts *m-unobservability* to quantify the uncertainty of associating POIs with a user's locations. According to [31], "unobservability is the state of Items of Interest (IOIs) being indistinguishable from any IOI at all." In the context of location-based services, IOIs are Points of Interest (POIs). Hence a user's location privacy can be defined as *unobservability* of his location (l). That is, the location is *m-unobservable* if and only if

$$H(l) = -\Sigma_{j=1}^{n} p(o_j) \log_2 p(o_j) \geq \log_2 m$$

where $o_j, j = 1, \ldots, n$ are the POIs and $p(o_j)$ is the probability that the user goes to o_j from his current location. We choose $\log_2 m$ as the threshold of information leakage (or entropy [34]) for m-unobservability because $\log_2 m$ is the

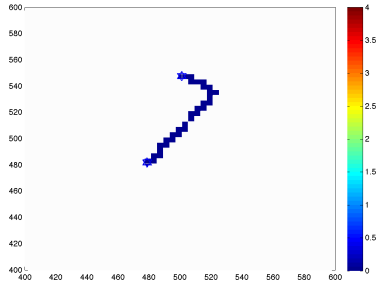


Fig. 2: Entropy map from the traces of one user in one day.

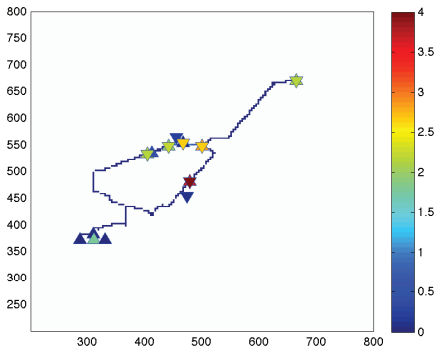


Fig. 3: Entropy map from the traces of all users in one day.

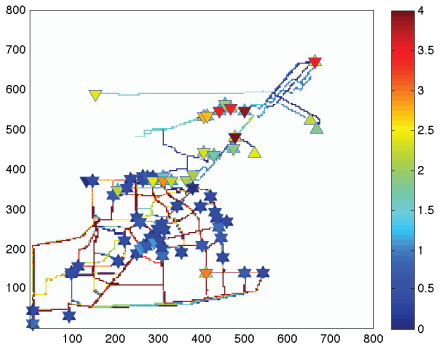


Fig. 4: Entropy map from the traces of all users in a week.

entropy value when the user is equally likely to visit one of m POIs. The intuition behind m-unobservability is that the adversary's ability to infer the user's private information is significantly weakened when they cannot relate any particular POI to the user's current location.

To derive $p(o_j)$, we divide a map area into a grid of 10m$\times$10m cells using pixelation similar to that in CacheCloak [26]. From users' location traces, each cell records the the number of times users go into the cell and finally, end at a particular POI, yielding n historical counters, $c_j, j = 1, \ldots, n$, where n is the number of POIs. Then $p(o_j)$ is defined as:

$$p(o_j) = \frac{c_j}{\sum_j c_j}. \quad (1)$$

For an illustrative purpose, we obtained the location traces of real users collected by Livelab [36] for over one year. Each log entry consists of a timestamp, the user's location in form of (latitude, longitude) and its accuracy. Collected separately is information about 600 POIs on the university campus. We select three subsets of traces: one user's traces on a certain day, eleven users' traces on the same day, and eleven users' traces of that week. For each data set, we calculate $p(o_j)$ for each POI using Eq. (1). The entropy of each cell is then used to color the cell on the map, as shown in Figs. 2, 3 and 4.

The unit of X and Y axes is the length of cell, 10m. The triangles indicate possible source and destination POIs and the line segments represent the roads connecting sources and destinations. The color bar on the right maps the entropy to the color temperature. Fig. 2 shows the entropy values calculated from two traces of a user going from one POI to another and coming back along the same road. With respect to each trace, the user only goes to one POI; therefore, the entropy values along the road are zero. Figs. 3 and 4 illustrate that, with more user traces and POIs, the entropy along the road increases. From a mobile user's perspective, the entropy map shows that location privacy, measured by unobservability, is affected by three fundamental factors: user activities, connectivity of roads, and location of POIs. Road segments that lead to few POIs have low entropy values and are thus privacy-revealing. On the contrary, those that are well connected often have high entropy values and are thus privacy-preserving because users may go through them to visit many different POIs.

V. Protection of Location Privacy

In this section, we describe the design goals of LISA and how it protects location privacy using m-Unobservability.

A. Design Goals

The design of LISA is steered by the following two goals.

- G1. *Privacy Protection*: it must prevent the distinguishability of *sensitive* POIs related to a mobile user, such that the attackers' ability to infer the user's privacy information is greatly weakened.
- G2. *Meeting Resource Constraints*: it must operate under various resource constraints imposed by smartphones, e.g. limited battery and processor capacity.

B. Overview of LISA

As mentioned in Section III, an LBS system consists of smartphones, wireless networks, and LBS servers. In each smartphone, `Query Composer` is responsible for sending location-service requests to, and receiving responses from, an LBS server. Suppose a user wishes to send her current location (x_0, y_0) and a query range $S = \{x, y|(x-x_0)^2+(y-y_0)^2 \leq L\}$ to an LBS server (for example, to find her friends or interesting events in her proximity defined by query range S). If she sends exact location (x_0, y_0), her privacy information can be inferred by attackers. To protect the location privacy, `Privacy Protection Engine` computes a "scrambled" location (x_0', y_0') and a larger query range $\S' = \{x, y|(x - x_0')^2 + (y - y_0')^2 \leq L'\}$ $(L < L')$, such that (x_0', y_0') satisfies the m-unobservability and the larger query range S' contains the original range S (this will be elaborated in the next section). Upon receiving the request from the user, the LBS server returns information inside S' (e.g. all her friends or events in S'), and then the smartphone performs local filtering, presenting users the information inside S.

C. Location Privacy Protection Engine

Fig. 5 illustrates how location information is scrambled by the local `Privacy Protection Engine` inside each smartphone. At the center of the protection engine, there are two Kalman filters to track users' movements based on a mobility model and scramble the user's location information to satisfy the m-unobservability requirement.

1) Mobility Model: To accurately track and predict a user's movements, the Engine uses the extended Kalman filter [28] which is often considered as the *de facto* standard for object tracking and navigation. We adopt the Wiener-sequence acceleration model [24] to model user' mobility, which assumes each acceleration increment is an independent (white noise) process. Let $\vec{x}_k$ denote the state variable and $\vec{y}_k$ the observation (a.k.a. measurement) variable at time t_k. In fact, the *process state*, $\vec{x}_k$, is a vector in the form of $(x, v_x, a_x, y, v_y, a_y)^T$, which represents the location, velocity, and acceleration of a user on the X and Y axes. The mobility model is given by

$$\vec{x}_{k+1} = A_k(\Delta t_k)\,\vec{x}_k + G(\Delta t_k)\,\vec{w}_k \quad (2)$$

$$\vec{y}_k = C\,\vec{x}_k + \vec{v}_k \quad (3)$$

where

$$A_k(\Delta t_k) = \begin{pmatrix} 1 & \Delta t_k & \Delta t_k^2/2 & 0 & 0 & 0 \\ 0 & 1 & \Delta\ t_k & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t_k & \Delta t_k^2/2 \\ 0 & 0 & 0 & 0 & 1 & \Delta\ t_k \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$G_k(\Delta t_k) = \begin{pmatrix} \Delta t_k^2/2 \\ \Delta t_k \\ 1 \\ \Delta t_k^2/2 \\ \Delta t_k \\ 1 \end{pmatrix} \qquad C = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}^T$$

$\vec{w}_k$ and $\vec{v}_k$ are both random variables for the process (system) noise and the observation (measurement) noise, respectively, and $\Delta t_k = t_{k+1} - t_k$. Based on this mobility model, the Kalman filter[1] can be defined as follows. The predicted process state is:

$$\hat{x}_{k+1|k} = A_k(\Delta t_k)\,\hat{x}_k \quad (4)$$

and the predicted estimate covariance is:

$$P_{k+1|k} = A_k(\Delta t_k)\,P_{k|k}\,A_k(\Delta t_k)' + Q_k \quad (5)$$

where the process noise covariance is:

$$Q_k(\Delta t_k) = cov(G_k(\Delta t_k)w_k) =$$

$$var(w_k) \begin{pmatrix} \Delta t_k^4/4 & \Delta t_k^3/2 & \Delta t_k^2/2 & 0 & 0 & 0 \\ \Delta t_k^3/2 & \Delta t_k^2 & \Delta t_k & 0 & 0 & 0 \\ \Delta t_k^2/2 & \Delta t_k & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta t_k^4/4 & \Delta t_k^3/2 & \Delta t_k^2/2 \\ 0 & 0 & 0 & \Delta t_k^3/2 & \Delta t_k^2 & \Delta t_k \\ 0 & 0 & 0 & \Delta t_k^2/2 & \Delta t_k & 1 \end{pmatrix}$$

For measurement updates, the Kalman gain is:

$$K_k = P_{k+1|k}\,C'\,[C\,P_{k+1|k}C' + R_{k+1}]^{-1},$$

the updated state estimate is:

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}[y_{k+1} - C\,\hat{x}_{k+1|k}]$$

[1]The filter is sometimes called *Stratonovich-Kalman-Bucy* filter because it is a special case of a more general, non-linear filter developed earlier by Stratonovich.

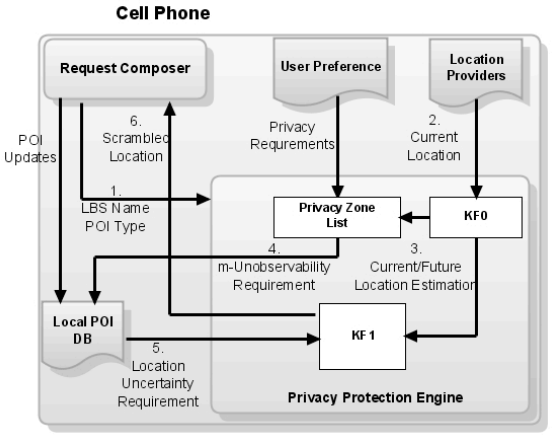


Fig. 5: `Privacy protection engine`

, and the updated estimate covariance is:

$$P_{k+1|k+1} = [I_6 - K_{k+1}\,C]\,P_{k+1|k}. \quad (6)$$

Next, we will discuss how this model is used to protect location privacy in the description of KF0 and KF1.

2) Two Kalman Filters: **KF0** denotes the extended Kalman filter that the Engine uses to track the user's movement. Such movement tracking is necessary for the following reasons. First, the Kalman filter can be used as an effective way to reduce the location measurement error, especially when users' location is determined by triangulation of radio towers[2]. Second, the `Protection Engine` uses accurate prediction of the user's movement (using Eq. (4)) to improve privacy protection and reduce resource consumption (a movement-prediction-based optimization is detailed in Section VI).

KF1 denotes the extended Kalman filter that the Engine uses to guide the scrambling of the user's location information. Given a location (x, y), a scrambled location, (X', Y') is a random variable following the normal distribution with mean $\mu = (x, y)$ and covariance Σ. In fact, Σ is the covariance of the measurement noise that the Engine injects to confuse attackers. From attackers' perspective, a scrambled location is a location measurement with some significant noise, which can be filtered or reduced by using mobility model. Hence, the role of KF1 is to examine a scrambled location and determine how much noise is needed to achieve the location uncertainty (given by Eq. (6)) that satisfies the privacy requirement. More specifically, `Privacy Protection Engine` scrambles a user's location information in following steps.

S1. When the user wishes to access an LBS, `Request Composer` tells the Engine to start generating a location that can be given to the LBS.

S2. The Engine obtains a location measurement from one of the location information providers, such as A-GPS or triangulation of radio towers. Assuming that the measurement is timestamped at t_k, the Engine updates the coefficient matrices of KF0

[2]Triangulation is still useful for smartphones without an integrated GPS (A-GPS) and for saving battery power even in phones with GPS [8]

and KF1 using $\Delta t_k = t_k - t_{k-1}$, and uses KF0 to estimate the current location $l_k = (x_k, y_k)$.

S3. With the current location l_k, the Engine looks up the entropy map and checks if the location satisfies the user's privacy requirement.

S4. If the location is m-unobservable, the Engine returns it to the Request Composer. Otherwise, go to S5 to perform location scrambling.

S5. Recall that R_k can be locally tweaked to affect $P_{k+1|k+1}$ by Eq. (6). Furthermore,

$$L_k \sim N(l_k, \begin{pmatrix} P_{k+1|k+1}(1,1) & 0 \\ 0 & P_{k+1|k+1}(4,4) \end{pmatrix})$$

, i.e., L_k is an unbiased estimation of l_k. The Engine gradually increases R_k so as to increase the covariance/uncertainty of L_k until $\bar{H}(L_k) = \Sigma_l g(l) * H(l) \geq \log_2 m$ is satisfied or the maximum noise level is reached.

S6. Once R_k is found, the Engine uses it to generate a random noise e_k by the normal distribution $N(0, R_k)$, and returns $l_k + e_k$ to the Composer.

VI. Optimizations

In this section, we propose two optimizations, look-ahead and de-randomization, to improve location-privacy protection.

A. Look-ahead

In Section V, we have showed that KF1's state-transition matrix, A_k, and process noise covariance, Q_k, are both determined by the inter-arrival time between consecutive LBS queries, Δt_k. As Δt_k gets smaller, A_k and Q_k both become smaller. As a result, the prior estimation error covariance $P_{k+1|k}$ approaches 0 and so does the Kalman filter gain, K_k. Then, the location measurement is trusted less, while the predicted location is trusted more. If $P_{k|k}$ is also small (i.e., not in high entropy location), increasing R_{k+1} cannot effectively increase $P_{k+1|k+1}$, and thus, cannot satisfy m-unobservability.

The look-ahead optimization alleviates this problem by predicting the location in the next step and adjusting the measurement noise in the current step accordingly in order to allow the next step to meet the unobservability requirements. To do so, the `Engine` predicts the location-privacy requirement in the next step by fixing the inter-arrival times of LBS queries, $\Delta t_k (k = 1, 2, \cdots)$ in Eq. (4) and use the predictd location to calculate its the entropy. Next, the `Engine` searches for the measurement noise covariances for the current step and the next step, R_{k+1} and R_{k+2}, such that both satisfy the privacy requirements and $R_{k+1} + R_{k+2}$ is minimized.

B. De-randomization

Another limitation in the previous section is that it does not consider user movement between two LBS queries. If a user stay in the same place but keeps sending LBS queries, his true locations may still be disclosed even though the queries only contain scrambled locations. The reason is simple: because the noises injected into the scrambled locations follow a 0 mean normal distribution, the average of the scrambled locations converges to the true location as the sequence gets longer. To address the problem, we propose to de-randomize the scrambled locations. Specifically, when the `Engine` detects less than minimum movement by a user, it reuses the previous scrambled location rather than generating a new one. This way, no new information can be obtained by attackers even as the mobile device continues to send LBS queries.

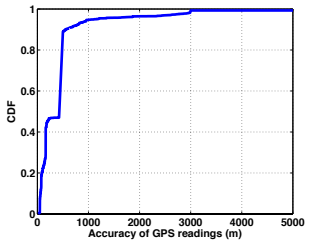


Fig. 6: GPS reading accuracy

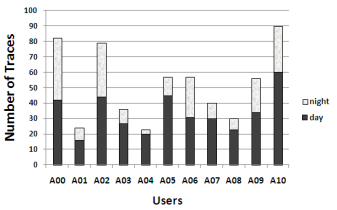


Fig. 7: User traces

VII. Evaluation

In this section, we evaluate the performance of LISA with real-world traces. First, we describe the traces used in the evaluation. Then, we present the experimental setup and performance metrics. Finally, we thoroughly analyze LISA's performance and explore the impact of varying settings on LISA's effectiveness in providing strong privacy protection for mobile users.

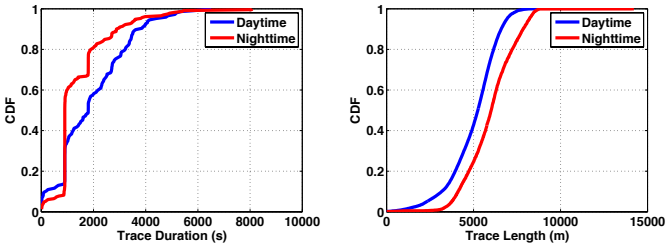
A. Real-world Traces

We use real-world traces of mobile users from the LiveLab Project at Rice University [36] that aims to measure long-term usage patterns of smartphones and wireless networks. To collect real user data, they recruited 25 participants from Rice University and gave each of them a smartphone equipped with the logging framework. The logged data was sent back to a centralized server on a daily basis, with sensitive personal information stripped off. The traces we obtained from LiveLab contain 11 users' GPS location readings over about one month. Each location log entry consists of the following six attributes: timestamp, latitude, longitude, accuracy of the location, altitude, and accuracy of the altitude. Overall, the raw data contains 45,151 GPS entries[3] covering a 409,915m $\times$ 89,502m area. The accuracy of GPS data varies significantly between 17m and 128km and its CDF (cumulative distribution function) is plotted in Fig. 6.

The initial step of processing the raw GPS data is to break down continuous GPS logs into movement segments (i.e., traces), representing users' trip from one POI to another. These segments can be used to create entropy maps and evaluate LISA's effectiveness. Based on the assumption that a user, after arriving at a particular POI, tends to stay at the same location, we use following process to separate user traces and identify the corresponding start/end POIs:

1) Filter GPS readings whose location inaccuracy is over 200m. The threshold is so chosen, balancing between the requirement for a sufficient amount of data and their quality (i.e., accuracy).
2) Cluster consecutive locations. Location entries are grouped such that the diameter of a group (i.e., the

[3]On average, 150 points were collected from each user every day.



(a) CDF: duration of user traces (b) CDF: length of the user traces

Fig. 8: Real-world trace statistics

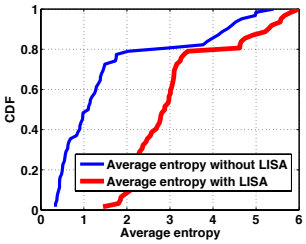


Fig. 9: CDF: Average location entropy of users with and without protection of LISA

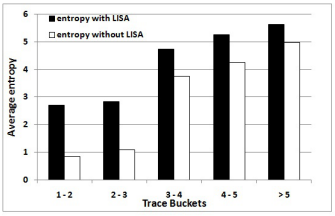


Fig. 10: Effectiveness of LISA for traces with different levels of location entropy

maximum distance between any pair of locations) is less than 500m. A "group" of locations indicate a small neighborhood/region.

3) Identify "stays" and "passings." If the duration of a group (i.e., the span from the earliest entry to the latest in a group) is more than 10 minutes, the user is considered to have stayed around the neighborhood. Otherwise, s/he is assumed to have passed it.
4) Identify segments of movement. If the time between consecutive groups (i.e., the earliest time in a group minus the latest time in its previous group) is over 60 minutes, then the two groups are assumed to be temporally unrelated and belong to two separate movement segments.
5) Construct a POI database. We compile a list of 650 POIs surrounding the Rice University campus including campus buildings, restaurants, shopping centers and other attractions.
6) Associate POIs with group of points. For each group of points at which users have stayed, we associate it with a POI in our POI database that has the minimum distance to all the points within the group.

The above algorithm has yielded 572 separate traces for all users over the entire monitoring period. Fig. 7 shows the number of traces collected from 11 users (noted as A00-A10) in different scenarios (i.e., day time vs. night time). From the figure, we can see that on average around 50 traces were collected from each user with the most active user contributing 90 traces over the five weeks. It is interesting to note that user activity patterns vary considerably across different users. Although users tend to be more active during the day time (372 traces) than during the night time (202 traces) and more active during weekdays (413 traces) than during the weekend (161 traces), some users exhibit different usage patterns. In particular, for A00, A01 and A06, the number of traces collected during the night is almost the same as (or even larger than) those collected during the day time.

Aggregating traces from all users, Figs. 8a and 8b show the CDFs (Cumulative Distribution Functions) of their duration (in seconds) and length (in meters) of day- and night-time traces. The empirical data show that users generally take short trips. For example, on average a trip takes about 30 minutes and the mean length of the trip is around 5km. Our data also shows that the trip length is shorter during the day than during the night and the trip duration exhibits the opposite trend. Overall the traces represent typical daily usage of smartphones and we use them to evaluate LISA in real-world settings

B. Experimental Methodology and Performance metrics

The GPS data from LiveLab was recorded at coarse timescale (every 15 minutes) due to high energy consumption of frequent location logging. For the evaluation of LISA's effectiveness, we refine these traces by feeding the original GPS records into VANETMobiSim [16] simulator to generate fine-grained location data along the coarse-grained traces. We first load into the simulator the map for area that covers most of the movement of the 11 users. Then, we use the staying and passing points of a GPS trace to define a user's trajectory and move virtual nodes intelligently between the points to create detailed location traces. It allows us to recover the real user's activity from coarse GPS logging data, which are then used to generate entropy maps and evaluate LISA's performance.

Specifically, two performance metrics are used to evaluate the effectiveness of privacy protection. The main metric is the location privacy entropy along the users' traces, which reflects an attacker's uncertainty in associating users with a particular POI. For example, 3 bits of entropy is equivalent to $2^3 = 8$ POIs equally likely to be related to a user location. However, it is difficult to set a fixed quantitative threshold for entropy, above which the user's privacy is considered protected and below which the user is identified. Thus, in the rest of this section, we will focus on the improvement on the location privacy LISA can achieve in different scenarios. Second, as LISA relies on injection of noises into location updates to protect the user's privacy, an important measure is the average spatial resolution provided by LISA. A lower spatial resolution (e.g. higher injected noises) means that LISA has to communicate more data with LBS servers and perform additional local filtering to remove irrelevant data points, leading to high energy costs. Therefore, LISA attempts to maximize the spatial resolution while maintaining sufficient location entropies.

C. Performance Evaluation

1) Location Entropy: We build entropy maps from first four weeks' user traces and use the fifth weeks' trace to evaluate LISA's performance in improving location entropies of the user paths. Fig. 9 shows the CDF of average entropies of users with and without LISA. From these results, we found that LISA performs especially well for low-entropy traces, often improving the location privacy by 2 bits (in many cases this is equivalent to a 200% increase in the entropy). Given that LISA stops noise injection if the location entropy reaches the entropy threshold (which is set to be $log_2 10 = 3.32$ in our

experiments), the improvement is significant. For high-entropy traces, the improvement is moderate. To demonstrate this effect, we place the test traces into buckets according to their average entropies. In total, we have five buckets holding traces whose entropies are between i and $i+1$ where $i = 1, 2, 3, 4$. For high-entropy traces with the entropy value greater than 5, we group them together in one bucket. Fig. 10 plots average improvement in different buckets using LISA. From the figure, we can see that LISA achieves, on average, 4.2 bits of entropy for all types of traces and 2.7 bits of entropy even for very low entropy traces. The average improvement for low entropy traces is about 1.7 bits, corresponding to a 180% increase.

D. Spatial Resolution

LISA improves the location privacy at the cost of increasing location uncertainties which inevitably reduces the spatial resolution and incurs additional communication and processing overheads. We measured this effect using the *relative temporal resolution* [12], which is defined as the temporal resolution provided by LISA normalized by the minimum acceptable temporal resolution (i.e., maximum tolerable bounding box size denoted as B_m). More specifically, let $\bar{B}$ represent the average bounding box size along the trace, the relative spatial resolution is defined as $B_m/\bar{B}$. A higher value implies more accurate location updates and thus a smaller communication overhead. Fig. 11 depicts the CDF of the relative spatial resolution, showing that in over 20% cases, the relative resolution is larger than 10 and in over 60%, the relative resolution is larger than 1. On average, the relative spatial resolution is 4.9, meaning that the bounding box generated by LISA is much smaller than the constraint box size. This demonstrates the effectiveness of LISA's noise tuning algorithm in generating the minimum possible bounding box to satisfy the privacy requirement.

E. Adaptive Entropy Map

Because a user's activity pattern may vary significantly between day time and night time, a potential way to improve LISA's performance is to adapt the entropy map to accommodate the dynamics of user mobility patterns. In particular, we create two separate entropy maps corresponding to user activities at day time and night time so that LISA can adaptively choose an appropriate entropy map to protect the user's privacy based on the local time. To evaluate the effectiveness of this adaptive entropy map, we use the same set of first four-week traces from all 11 users and separate them into two types of traces, namely, day-time and night-time traces. Each set is used to create one entropy map. Then, we also separate the traces of the fifth week into these two types and run them through LISA to collect results about average entropies of user traces. The results, summarized in Fig. 12, are compared using the same bucketing algorithms to better demonstrate the advantages of the adaptive approach. First, we observe that for low-entropy traces, using the adaptive entropy map allows LISA to achieve higher location entropy in both day and night times than the combined entropy map. For example, the average entropy for the first bucket (1-2) is 2.98 for day time, 2.74 for night time and 2.70 for combined entropy maps. The difference is even larger for the second bucket. On the other hand, for high-entropy traces, using adaptive entropy maps leads to a smaller bounding box size and reduced the communication overhead. For example, the average bounding box size for day-time entropy for the fourth bucket (4-5) is 79.6m compared to the 179.6m bounding box for the combined entropy map. With fewer user activities in the night, the night time entropy map yields a bounding box size of 210m, which is justifiable since user entropies tend to be low in the night. Although for high-entropy traces, the adaptive entropy map appears to reduce average location entropy below the combined entropy map. However, the location entropy is still above the threshold (i.e., $log_2 10$)) and provides a desirable level of location privacy. Trading this additional entropy for a smaller bounding box size, LISA reduces the communication and processing overheads without failing to meet the required privacy for mobile users.

F. Cross Validation Among Users

One potential reason for LISA to achieve good performance could be due to the temporal correlation of traces from the same person. Therefore, to show LISA's robustness to the user population, we perform cross validation with different sets of users. Specifically, we create an entropy map using traces from 10 out of total 11 users (i.e. training traces), and then use the traces of the remaining user as the "test" traces for LISA. We performed the cross validation for all 11 cases and summarized the results in Fig. 13. In Fig. 13, each bar represents the location entropy averaged across 11 different combinations and the error bars indicate the minimum and maximum values. Although there are variations in the amount of entropies LISA can provide (likely caused by traces that is unique to individual users), LISA still achieves high privacy protection, in particular for low-entropy traces. For example, for low-entropy traces in the first bucket, LISA, on average, increases its location entropy by 2.3 bits, resulting in an entropy value that is close to the threshold ($log_2 10$) and almost 3 times more than achieved without LISA. This result suggests that after accruing a relatively large number of user traces, LISA can be generalized to previously unseen users.

VIII. Conclusion

We have proposed a new approach, called the *Location Information ScrAmbler* (LISA), to protecting the location privacy of mobile users. The key idea of LISA is *m-unobservability*, which disables the distinguishability of POIs a user may visit, and therefore, weakens the attackers' capability of inferring his privacy information. Based on a simple mobility model, LISA achieves protection of location privacy and resource efficiency by tweaking the noise used to scramble the locations revealed to LBSs. LISA introduces a new, orthogonal (relative to others) dimension of uncertainty, and can be combined with other approaches to enhance location privacy. Moreover, it eliminates the need for trusted third-party servers, thus lowering the risk of attacks and simplifying the implementation and deployment of LBS systems. We evaluated the performance of LISA using real-world traces. The results show that LISA can effectively protect location privacy with good resource efficiency.

Acknowledgements

The work reported in this paper was supported in part by the NSF under Grant CNS-1114837 and the ARO under Grant W811NF-12-1-0530.

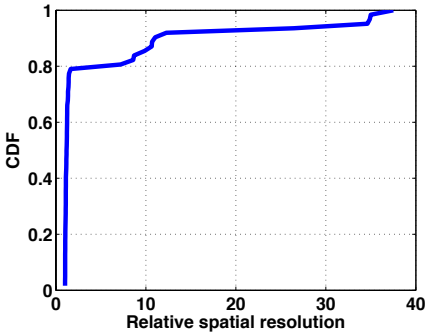


Fig. 11: CDF: Relative spatial resolution

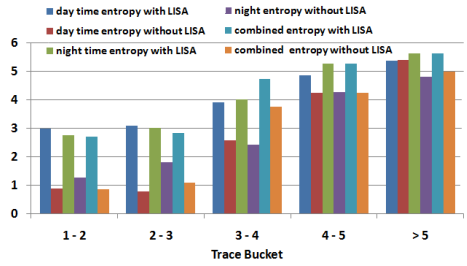


Fig. 12: Comparison of LISA performance under day-time, night-time and combined entropy maps

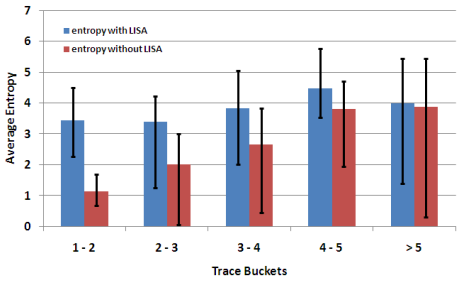


Fig. 13: Cross-validation results of LISA's performance on different traces

REFERENCES

[1] Al Franken. Location privacy protection act. http://beta.congress.gov/bill/112th-congress/senate-bill/1223, 2012.

[2] Alohar. Placeme. https://www.placemeapp.com/placeme/, 2011.

[3] Alohar Inc. Alohar. http://techcrunch.com/2012/04/17/alohar-mobile-helps-developers-build-smarter-apps/, 2012.

[4] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. In *Pervasive Computing*, 2003.

[5] A. Bose, X. Hu, K. G. Shin, and T. Park. Behavioral detection of malware on mobile handsets. In *Proceedings of the ACM/USENIX MobiSys*, 2008.

[6] Citysense Inc. Citysense. http://www.citysense.com.

[7] Computer Science and Telecommunications Board. It roadmap to a geospatial future, Nov 2003.

[8] I. Constandanche, M. Sayler, S. Gaonkar, R. R. Choudhury, and L. Cox. Enloc: Energy efficient localization for mobile phones. In *Infocom*, 2009.

[9] A. Deutsch, R. Hull, A. Vyas, and K. Zhao. Policy-aware sender anonymity in location based services. In *ICDE*, March 2010.

[10] Electronic Frontier Foundation. 2012 in review: Major location privacy developments. https://www.eff.org/deeplinks/2012/12/2012-review-major-location-privacy-developments, 2012.

[11] Facebook. Find friends nearby. http://techcrunch.com/2012/06/24/friendshake-facebooks-new-mobile-feature-for-finding-people-nearby-and-a-highlight-killer/, 2012.

[12] B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *ICDCS*, 2005.

[13] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan. Private queries in location based services: anonymizers are not necessary. In *SIGMOD*, pages 121–132. ACM, 2008.

[14] M. Gruteser and D. Grunwala. Anonymous usage of location-based services through spatial and temporal cloaking. In *ACM/USENIX MobiSys*, pages 31–42, 2003.

[15] M. Gruteser and B. Hoh. On the anonymity of periodic location samples. In *Security in Pervasive Computing*, 2005.

[16] J. Härri, F. Filali, C. Bonnet, and M. Fiore. Vanetmobisim: generating realistic mobility patterns for vanets. In *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, 2006.

[17] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *SecureComm*, 2005.

[18] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J. C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *MobiSys*, 2008.

[19] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Enhancing security and privacy in traffic-monitoring systems. In *Pervasive Computing*, 2006.

[20] H. Hu and J. Xu. Non-exposure location anonymity. In *ICDE*, pages 1120–1131. IEEE, 2009.

[21] T. Jiang, H. J. Wang, and Y.-C. Hu. Preserving location privacy in wireless lans. In *Mobisys*, 2007.

[22] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *ICPS '05. Proceedings. International Conference on Pervasive Services*, 2005.

[23] J. Krumm. Inference attacks on location tracks. In *Pervasive Computing*, 2007.

[24] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39:1333– 1364, 2004.

[25] Y. Matsuo, N. Okazaki, K. Izumi, Y. Nakamura, T. Nishimura, and K. Hasida. Inferring long-term user properties based on users location history. In *IJCAI*, 2007.

[26] J. Meyerowitz and R. R. Choudhury. Hiding stars with fireworks: Location privacy through camouflage. In *the 15th Annual International Conference on Mobile Computing and Networking*, 2009.

[27] Mobile Internet Capital Inc. Our investment targets. http://www.mickk.com/en/eclassify1.html.

[28] T. K. Moon and W. C. Striling. *Mathematical Methods and Algorithms for Signal Processing*. Prentice-Hall, 2000.

[29] NextBus Inc. Nextbus. http://www.nextbus.com.

[30] S. Papadopoulos, S. Bakiras, and D. Papadias. Nearest neighbor search with strong location privacy. In *VLDB*, September 2010.

[31] A. Pfitzmann and M. Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management - a consolidated proposal for terminology. http://dud.inf.tu-dresden.de/Anon_Terminology.shtml, Feb. 2008. v0.31.

[32] A. Pingley, W. Yu, N. Zhang, X. Fu, and W. Zhao. Cap: A context-aware privacy protection system for location-based services. In *ICDCS*, pages 49–57. IEEE, June 2009.

[33] A. Pingley, N. Zhang, X. Fu, H.-A. Choi, S. Subramaniam, and W. Zhao. Protection of query privacy for continuous location based services. In *INFOCOM'11*. IEEE, April 2011.

[34] C. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.

[35] A. Sharma and J. E. Vascellaro. Phones will soon tell where you are. http://online.wsj.com/article/SB120666235472370235.html, 2008.

[36] C. Shepard, A. Rahmati, C. Tossell, L. Zhong, and P. Kortum. Livelab: Measuring wireless networks and smartphone users in the field. In *HotMetrics*, 2010.

[37] R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux. Quantifying location privacy. In *IEEE Symposium on Security and Privacy*. IEEE, May 2011.

[38] The Economist Magazine, Special Report, April, 2008. Location, location, location it matters.

[39] W3C. Platform for privacy preferences (p3p) project. http://www.w3.org/P3P, Retrieved April 2011.

[40] T. Xu and Y. Cai. Feeling-based location privacy protection for location-based services. In *CCS*, pages 348–357. ACM, November 2009.