Fingerprinting Mobile User Positions in Sensor Networks: Attacks and Countermeasures

Mo Li, *Member*, *IEEE*, Xiaoye (Jonathan) Jiang, *Student Member*, *IEEE*, and Leonidas J. Guibas, *Fellow*, *IEEE*

Abstract—We demonstrate that the network flux over the sensor network provides fingerprint information about the mobile users within the field. Such information is exoteric in the physical space and easy to access through passive sniffing. We present a theoretical model to abstract the network flux according to the statuses of mobile users. We fit the theoretical model with the network flux measurements through Nonlinear Least Squares (NLS) and develop an algorithm that iteratively approaches the NLS solution by Sequential Monte Carlo Estimation. With sparse measurements of the flux information at individual sensor nodes, we show that it is easy to identify the mobile users within the network and instantly track their movements without breaking into the details of the communicational packets. Our study indicates that most of existing systems are vulnerable to such attack against the privacy of mobile users. We further propose a set of countermeasures that redistribute and reshape the network traffic to preserve the location privacy of mobile users. With a trace driven simulation, we demonstrate the substantial threats of the attacks and the effectiveness of the proposed countermeasures.

Index Terms—Sensor networks, network flux, mobile user, fingerprint.

1 Introduction

Recent advances in Wireless Sensor Network (WSN) technologies envision more pervasive usage of the sensor network where the human beings are deeply interacting with the cyber-physical environment. In addition to the traditional paradigm of data collection from remote sensor networks, people may coexist in the same physical space of interest with the sensor network infrastructures. Equipped with 802.15.4 compatible communicating devices, each user is able to move around within the sensor network and directly communicate with nearby sensors, capable of pervasive access to the instant data over the entire field.

In such a pervasive context of data access, the deployed infrastructural sensor network is capable of simultaneously supporting multiple mobile users and providing them with field data in an anyone-anywhere-anytime manner. There have been substantial applications based on this data access mechanism, from ubiquitous data acquisition to human navigation, etc., [11], [12]. The mobile users access the network at different locations and acquire network-wide data instantly through intermediate nodes.

In this paper, however, we demonstrate that such a working paradigm suffers from a potential risk of leaking the location privacy of users. With alarming ease, a malicious entity can track the every move of mobile users only from passively sniffing the network traffic flux at a sparse set of points. They do not even need to break into the content of data packets.

The mobile users access the network at different locations and produce their own traffic flows, respectively, across the network. In most of existing works, a data collection tree is built for each mobile user and network-wide data are delivered by intermediate sensor nodes along the tree [9], [13]. The produced traffic flows of different mobile users add upon each other at intermediate nodes and the traffic amounts cumulate. If we summarize the traffic flux distributed over the network we get a flux pattern of particular shape. Fig. 1 depicts the network flux pattern where there are three mobile users collecting data from the network. Fig. 1a presents the three mobile users and their data collection trees built across the network and Fig. 1b depicts the network flux pattern introduced by the mobile users. Indeed, the pattern of the network flux is related to the statuses of mobile users. It digests the information including the number of mobile users, their locations, their traffic stretches, etc. Thus, by exploring the traffic pattern over the network, the adversaries are able to build a mapping between the instant distribution of mobile users and the observed network flux.

As reported in our preliminary work [10], a parameterized model is built to abstract the network flux with different situations of mobile users. By fitting the theoretical model to the measurements on real network flux, we are able to gradually identify the locations of mobile users distributed over the field. While gathering the flux information over the entire network might be of heavy overhead, we show that even with sparse measurements of the flux at a small set of individual sensor nodes we are still able to fingerprint the mobile users through

- *M. Li is with the School of Computer Engineering, Nanyang Technological University, N4-02c-108, 50 Nanyang Avenue, Singapore 639798. E-mail: limo@ntu.edu.sg.*
- *X. Jiang and L.J. Guibas are with the Computer Science Department, Stanford University, Gates Building, Stanford, CA 94305. E-mail: xiaoyej@stanford.edu, guibas@cs.stanford.edu.*

Manuscript received 19 Oct. 2010; revised 9 June 2011; accepted 27 June 2011; published online 21 July 2011.
Recommended for acceptance by C. Pinotti.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2010-10-0620.
Digital Object Identifier no. 10.1109/TPDS.2011.213.

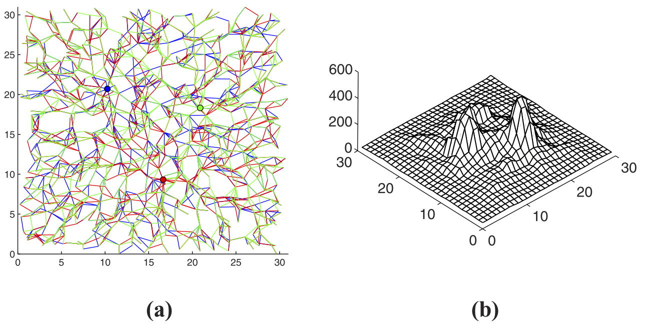

Fig. 1. The network flux with three mobile users. (a) The data collection trees. (b) The network flux pattern.

parameter fitting. We further develop an algorithm that iteratively approaches the movements of mobile users by Sequential Monte Carlo Estimation technique. As more flux measurements are cumulated, our algorithm converges to the moving trajectories of mobile users and approximates their locations with high accuracy.

A particular threat of such an attack is that compared to the vast information that can be revealed, the required knowledge is extremely cheap. Only sparse knowledge of the network flux is enough for the entire calculation. As a matter of fact, due to the broadcast nature of the wireless communication medium, such information is easy to access through passive sniffing and the malicious entities could exchange such sniffing information over external channels without the network nodes' awareness. As a direct result, we demonstrate that most existing systems are vulnerable in protecting the privacy of mobile users. With respect to the preliminary conference version, in this paper, we further propose a set of countermeasures to redistribute and reshape the network traffic. By considering the traffic distribution and moving trajectories of mobile users in building the data collection tree, we are able to smartly smoothen the traffic pattern without imposing much extra traffic burden to the network. We can, thus, prevent leaking the location privacy with little extra overhead. We extensively evaluate the proposed countermeasures with real data trace driven simulations. The results in Appendix H, which can be found on the Computer Society Digital Library at http://doi.ieeecompute rsociety.org/10.1109/TPDS.2011.213, suggest that the traffic redistributing and reshaping are effective and efficiency in addressing such potential attacks.

The rest of the paper is organized as follows. Section 2 briefly introduces related work. Section 3 describes the main design rationale. In Section 4, we give detailed descriptions on how we fingerprint the mobile users with sparse samplings of network flux. We propose a set of countermeasures against the attack in Section 5. In Section 6, we validate our design with extensive simulations. Finally, we conclude this work in Section 7.

2 Related Work

Knowing accurate locations of interesting objects or people is of essential importance for many pervasive applications. Initial attempts of the research community include LANDMARC [15], RADAR [2], Cricket [17], etc. There have been also many approaches proposed for locating and tracking objects within the sensor network. More details about those related works can be found in Appendix A, available in the online supplemental material, [6], [8], [17], [18]. Different from all existing studies, in this work, we demonstrate that even when both the moving entities and the sensor network infrastructures are noncooperative, a malicious entity can still identify the mobile users with minimum information that is difficult to secure.

The problem of disclosing user privacy in wireless network context has recently drawn the concern of research community. There have been studies showing that the location privacy could be vulnerable with the "broadcast" wireless communication channels [1], [3], [16]. They demonstrate that the adversaries are able to acquire user locations with wireless fingerprint information that can be obtained through direct or indirect access to the inbound and outbound traffic nearby the user. Most existing studies assume that the adversaries have direct access to the data packets or heavy monitoring of the traffic flows to obtain necessary fingerprint information. In this work, however, we show that a sparse sampling on the amount of traffic flux in the field suffices to reveal fair amount of location privacy of mobile users, which is much cheaper and easier for the malicious entities to launch. Deng et al. [5] noticed similar attacks based on traffic analysis. Their work, however, focuses on the scenario where there is only one static base station within the network. There is also no detailed analysis on how the adversaries can absorb location information from the traffic patterns.

3 Rationale

The goal of adversaries is to solely utilize network flux information to fingerprint the mobile users within the sensor network field. In this section, we first formalize the problem that we are studying, including the application scenario, design objective, assumptions, etc. We then develop a parameterized model to predict the network flux over the field. We introduce the basic design rationale of locating mobile users through briefing the network flux.

3.1 Problem Statement

We consider the scenario where multiple mobile users move around within the sensor network field, collecting the sensory data from the network.

Let the number of mobile users be K. Each mobile user repeatedly collects the updated data from the network at its own will. The data collection of each user happens at different time and different places. For any mobile user i, there exists a time series of data collections $[t_1^i, t_2^i, \ldots, t_{ki}^i]$ while the corresponding positions are $[p_1^i, p_2^i, \ldots, p_{ki}^i]$. Different users may have different time series of data collections independent of each other. Our goal is to track those mobile users, i.e., to figure out the location instances of each mobile user $\{[p_1^i, p_2^i, \ldots, p_{ki}^i] | 1 \leq i \leq K\}$.

Toward such a goal, minimum capability is required by the adversary to perform attack on location privacy. What we assume available for the adversary are the instant measurements of the traffic flux over the network. The adversary launches *outside* attack in the network, i.e., the adversary does not have to compromise any legitimate nodes yet does not need to break into the content of any data packet delivered over the network. Such *outside* attack, however, is particularly dangerous as the scope of the attack is broad. Even in the sensor network with highly secured communicational links, where the internal network operations can be protected by light-weight symmetric-key or asymmetric cryptographic mechanisms, the adversary can obtain the desired traffic flux information by simply sniffing the data amount delivered on the air.

We assume that the adversary measures the network traffic flux at each time window ΔT. The time window ΔT determines the measurement granularity. When $\Delta T \rightarrow 0$, we get ever more delicate observation of the network flux. In practice, ΔT is limited by the inherent duration of wireless transmissions, synchronization among different observers, etc. Nevertheless, with current technologies, ΔT can be bounded at the "seconds" level, leading to minor observation error compared with the intrinsic system error brought by the discrete position estimations with "minutes" intervals. Within each time window, different mobile users may or may not happen to initiate the data collection. In a more general way as adopted in most existing works, when one mobile user wants to collect the data from the network, it builds a data collecting tree that roots at the sink and spans the network. Different mobile users may have different traffic stretches, i.e., they collect different proportions of data from each node due to their interests at different environment aspects. The measured network flux at each time window is the sum-up of the traffic F_i initiated by each mobile sink. At each node, we can measure the cumulated traffic flux

$$F = \sum_{i=1}^{K} F_i.$$

However, we cannot exactly separate each share of the flux amount introduced by each mobile user. Instead, we develop a mathematical model to fit the mobile user statuses according to such combined fingerprint flux information.

3.2 Network Flux Model

In this section, we study how the network flux is composed when the mobile user absorbs data from the network-wide data collecting tree. We accordingly build a network flux model to approximate the amount of data flux at each node.

Note that the data flux at each intermediate node is the cumulated amount of data it generates and relays, including the data generated at all successor nodes on the subtree it roots. We first consider a continuous scenario where sensor nodes are deployed over the field with infinite density. Fig. 2a depicts a sector-like region of angle ω and radius l originated at the user. We assume that each point within the sector-like region generates a unit of data and the traffic stretch is s for each unit area. For the arc a which is d distant from the sink, all data generated at points beyond a (in the blue area) pass the arc. Let the average traffic flux at each point on arc a be F_a. We have the entire amount of data delivered across a

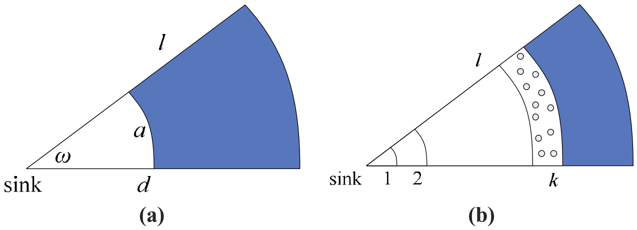

Fig. 2. Illustration of the network flux model. (a) Continuous case. (b) Discrete case.

$$M_a = \int_{\theta=0}^{\omega} \int_{d}^{l} s \cdot r dr d\theta = \int_{\hat{a}} F_a d(x, y). \tag{3.1}$$

From (3.1), we get $F_a = s(l^2 - d^2)/2d$, which is independent of the angle ω. We let $\omega \rightarrow 0$ and obtain that the flux at each intermediate point F_d is determined by the distance d from the sink to that point and the distance l from the sink to the network boundary along the direction of that point

$$F_d = s(l^2 - d^2)/2d. \tag{3.2}$$

Formula 3.2 models the traffic flux for the ideal network of infinite node density. For a more practical flux model, we further generalize our analysis for the discrete networks. Fig. 2b illustrates how the data flux concentrates at the k-hop away nodes from the user. All k-hop nodes reside inside the strip area k hop distant from the user and all nodes beyond k hop away from the user (in the blue area) have their data amount relayed by those k-hop nodes. Let the flux at each k-hop node be F_k. We have the entire amount of data transmitted through those k-hop nodes is

$$M_k = \frac{\omega(k^2 r^2 - (k-1)^2 r^2)}{2} \cdot \rho \cdot F_k = \frac{\omega(l^2 - (k-1)^2 r^2)}{2} \cdot \rho \cdot s, \tag{3.3}$$

where r is the average distance of each hop, ρ is the node density, and s is the traffic stretch of the current sink.

From (3.3), we get $F_k = s(l^2 - (k-1)^2 r^2)/(2k-1)r^2$. We can reformulate it and approximate F_k with items of real distance variables

$$F_d \approx s(l^2 - d^2)/2dr. \tag{3.4}$$

Indeed, Formula 3.4 is a consistent representation of Formula 3.2 in the discrete setting with a division factor of the average hop distance r. According to Formula 3.4, the position of the mobile user determines the parameters l and d, thus affecting the traffic flux at intermediate nodes. Using Formula 3.4, we are able to approximate the traffic flux at any position for general discrete networks. On the other hand, if we have the measurements of the traffic flux at each node, Formula 3.4 allows us to identify the location of the mobile user with a parameter fitting. Indeed, if we average the amount of flux within the neighborhood of an intermediate node, we are able to get a smoother map of the network flux and better approximation accuracy by mitigating the randomness of routing tree construction. In Appendix B, available in the online supplemental material, more statistical results on the accuracy of the network flux model is presented.

3.3 Briefing the Network Flux

According to the analysis in the previous section, with the summary of traffic flux over the network we are able to identify the location of the mobile user by simply extracting the point of traffic concentration. The problem, however, becomes a bit more difficult when there are multiple mobile users initiating data collection at the same time. As Fig. 1 demonstrates, there are three mobile users collecting network data with three different data collecting trees and their traffics cumulate at intermediate nodes. Under such circumstances, simply detecting the traffic peaks is not effective any more. It is difficult to distinguish traffics of different mobile users and the traffic flux of one mobile user may heavily influence the observation on other mobile users, especially when different mobile users have different traffic stretches.

To address such a problem, we use a recursive method to brief the observed network flux with our theoretical model. We identify the positions of mobile users in multiple rounds. In each round, we detect the global traffic peak and accordingly identify the position of a mobile user. We then estimate the traffic stretch of the mobile user with the peak traffic. With the theoretical model, we are, thus, able to approximate the network traffic flux associated with the current mobile user. We then deduct the associated traffic amount over the field from the originally observed network flux map. By such a method, at each round, we can always identify the location of a mobile user with dominating traffic flux and then get the corresponding traffic deducted from the network flux map. In Appendix C, available in the online supplemental material, we demonstrate how the method works for the example of Fig. 1.

Such a method, however, requires the flux information over the network to capture the desired traffic peaks. Such a requirement leads to expensive operational overhead, i.e., sniffing all the nodes within the entire network field. Nevertheless, in next section, we show that we can fingerprint the mobile users with only sparse samplings of the network flux, significantly reducing the overhead of launching the attack.

4 Fingerprinting with Sparse Samplings

Instead of acquiring the traffic flux information over the entire network, we can merely use sparse samplings on a small portion of nodes to get adequate fingerprint information. We do a parameter fitting on our theoretical flux model according to the node flux samplings over the network such that we can find the best possible distribution of mobile users. We further develop an algorithm that iteratively approaches the mobile sink movements by Sequential Monte Carlo Estimation technique.

4.1 NLS Parameter Fitting

With only sparse samplings from a small portion of nodes, we are not able to directly map those traffic peaks of mobile users. Instead, we do parameter fitting on our theoretical flux model such that the flux measurements can be the best fit.

Assume that we have the flux samplings at n nodes which are evenly distributed across the field. From the theoretical model indicated by Formula 3.4, we can estimate the flux vector F at sampling nodes and compare with the real measurement vector F'. The best parameter fitting corresponds to a Nonlinear Least Squares (NLS) optimization problem, which will minimize the following objective function:

$$\min.\|F - F'\|$$
$$\begin{cases} F_i = \sum_{j=1}^{K} \frac{s_j}{r} \cdot \frac{(l_{i,j}^2 - d_{i,j}^2)}{2d_{i,j}}, (i = 1, 2, \dots, n), \\ l_{i,j} = l(x_i, y_i, x_j, y_j), \\ d_{i,j} = d(x_i, y_i, x_j, y_j). \end{cases} \tag{4.1}$$

Here, F_i describes the estimated flux amount at the ith node according to the flux model, where the traffic of K mobile users cumulates. The estimated value F_i is determined by the positions of mobile users (x_j, y_j), and their traffic stretches s_j. We try to fix such parameters as the solution $\in \Re^{3K}$ for this optimization problem. Indeed, the number of mobile users K is not necessarily preknown. For the cases where we do not know the exact number of mobile users in the field, we can conservatively choose a K large enough, and after the optimization process the K coordinates will converge at the actual positions of mobile users. There is an unknown constant r in the function which measures the average distance of each communication hop. In practice, r is limited by the maximum communication radius R, but is different under different network densities. Nevertheless, we take s_j/r as an integrated factor and fit its value. Directly applying numerical techniques to solve the above NLS problem is not feasible in some situations, where the objective function may not be differentiable.

As a matter of fact, the shape of the network boundary determines our function of calculating $l_{i,j}$. A nondifferentiable network boundary, say, a rectangular field, usually leads to the nondifferentiable objective function. Traditional numerical techniques like the Gauss-Newton method or the Levenberg-Marquardt method [14] all require the objective function to be differentiable, thus not applicable in those cases. On the other hand, the direct solution of the NLS problem is not always a stable estimation of the locations of mobile users, due to the measurement errors and model prediction errors. The estimated locations may largely vary between consecutive estimations with different instances of flux observations.

Against such challenges, we propose to approach the mobile user movement with sequential samplings. Under the NLS constraints, we can efficiently filter those outlier samplings and keep a good approximation. With the Sequential Monte Carlo Sampling technique, we are able to cumulate our prior observations on network flux and get constantly refined estimation accuracy.

4.2 Sequential Monte Carlo Estimation

In our problem, for each mobile user i, there exists a time series of data collections $[t_1^i, t_2^i, \dots, t_{ki}^i]$ corresponding to the sequence of its positions $[p_1^i, p_2^i, \dots, p_{ki}^i]$. The Sequential Monte Carlo method allows us to represent the real position of the mobile user p_j^i at each instance j with a set of random samples. Those samples are updated iteratively with the importance sampling method. Through the prediction and filtering operations in each round of update, we are able to restrict the samples to the posterior distribution of the mobile user's possible positions.

Let t be the discrete time instances. For mobile user i, it corresponds to the time series of data collections $[t_1^i, t_2^i, \ldots, t_{ki}^i]$. Let p_t represent the position distribution at time t. We can predict the current position distribution of the mobile user from its previous position, i.e., $P(p_t|p_{t-1})$. On the other hand, according to our observations on the network flux we get the likelihood of the mobile user's current position with the observation constraints, i.e., $P(p_t|o_t)$. With sequential observations on the network flux evolutions, we iteratively approach the posterior distribution $P(p_t|o_1, o_2, \ldots, o_t)$. At each stage, we use a set of N random samples P_t to approximate the position distribution p_t. We accordingly update the set of samples as the observed network flux pattern evolves. At each time instance t, P_t is computed with the previous approximation P_{t-1} and the current observation o_t.

4.3 Prediction and Filtering

Initially, without any knowledge and constraints on the position of the mobile user we assume a uniform distribution and select the samples uniformly random over the field. At each time step, we predict the possible positions of the mobile sink based on the transition distribution $P(p_t|p_{t-1})$ and get updated position samples. We then eliminate those predictive samples inconsistent with network flux observations in a filtering phase. In such a process, the sampling distribution gradually approaches the posterior distribution $P(p_t|o_1, o_2, \ldots, o_t)$.

In the prediction phase, we get the updated set of samples P_t from the previous set P_{t-1}. We assume a weak model to predict the movement of the mobile user, i.e., we do not have any specific information on its mobility pattern (speed, direction, trajectory, etc.) except the knowledge of its maximum moving speed v_{max}. Thus, from any sample position in the previous step $P_{t-1}(i)$, the possible current position $P_t(i)$ is uniform random within a circular region of radius $v_{max} \cdot \Delta t$, where Δt is the time interval between the two consecutive time instances

$$P(p_t|p_{t-1}) = \begin{cases} \dfrac{1}{\pi(v_{\max} \cdot \Delta t)^2}, & if\ d(p_t, p_{t-1}) \leq v_{\max} \cdot \Delta t \\ 0, & if\ d(p_t, p_{t-1}) > v_{\max} \cdot \Delta t. \end{cases} \tag{4.2}$$

After the prediction phase, there are N new samples drawn randomly from the discs centered at previous sample origins, corresponding to increased uncertainty on the movement of the mobile user. Indeed, above mobility model can be further refined if we have more accurate mobility prediction, say, the heading of the mobile user.

In the filtering phase, we eliminate those impossible position samples from P_t to cut down the uncertainty due to the unawareness of mobility. The filtering operation is bound to our network flux observations. For each mobile user i, we estimate the incurred network flux when it is at any of the N possible updated positions. We sum up the flux amounts incurred by all K mobile users and obtain the estimated flux vector F for the n sampling nodes. For all N^K possible combinations of the mobile user positions, we estimate the flux vector F and compare it with the real measurement F'. Since there still exists freedom on the traffic stretches of mobile users, we take $s_j/r (j = 1, 2, \ldots, K)$ as integrated factors and fit their values to minimize $\|F - F'\|$. We are then able to find minimized objective value $\|F - F'\|$ for each possible combination of the mobile user positions, with specific traffic stretch factor s_j/r. Such observations allow us to filter out those position combinations apart from real measurements by their objective values. We rank the N possible updated positions for each mobile user i, according to their minimum objective values each of which is achieved in N^{K-1} possible combinations. Finally, we keep the top M updated positions for each mobile user and filter out the other possible positions.

4.4 Asynchronous Updating

Recall that in our application context different mobile users collect the updated data from the network at their own wills. For any mobile sink i, there exists a time series of data collections $[t_1^i, t_2^i, \ldots, t_{ki}^i]$ which is independent with each other. As a matter of fact, the observable updating of their positions is by nature asynchronous. For each round of observing the network flux some mobile users may not happen to collect data from the network and there is a best fit traffic stretch $s_j/r \rightarrow 0$ estimated for each of them in the prediction and filtering phase. In such a case, we will not update the position samples of those mobile users and instead we allow a larger Δt for computing the transition distribution $P(p_t|p_{t-1})$ in following rounds. As a result, the samples of different mobile users are asynchronously updated. For each mobile user i, the time interval Δt used to calculate the movement radius $v_{max} \cdot \Delta t$ in Formula 4.2 is the time period between two consecutive time points of data collection $t_j^i - t_{j-1}^i$.

We can further improve the estimation process with importance sampling and we give the details in Appendix D, available in the online supplemental material. The pseudocode of the Sequential Monte Carlo Estimation is shown in Appendix E, available in the online supplemental material.

5 Countermeasures

As the previous section demonstrates, with current data collection style, the mobile users take high risk of leaking their location privacy to malicious parties. In this section, we further discuss such a problem and propose a set of countermeasures to secure the location privacy of mobile users.

5.1 Redistributing Network Traffic Flux

As the major source that exposes the locations of mobile users is the network flux incurred during the data collection, we change the data collection style and redistribute the net work traffic flux. To redistribute the traffic flux over the network, a straightforward idea is to let sensor nodes distribute fake traffics across the network which are meaningless to the application, but add extra traffic amounts on top of existing traffic flux. In such a way, we are able to hide the real traffic patterns with extra traffic jamming from the malicious entities. However, the extra traffic introduced by such a method is excessively high, which will become a large overhead to the sensor network and may overwhelm the regular operations, leading to extremely low networking efficiency. In order to neatly address such an issue, we propose a light-weight approaches that significantly smoothen the network traffic while maintain low extra traffic amount.

In existing method to build data collection trees, the hop count is the sole metric used in the construction process. Such data collection trees are optimal in terms of path length, i.e., each sensor node is connected through a shortest path of minimum hop count to the mobile user, but lead to apparent traffic flux distribution that can be captured by the adversary party. In our scheme, when building the data collection tree we let each sensor node take both hop count metric and traffic flux metric into consider. An arbitrary node in the network receives control messages from its neighboring nodes, containing hop count indicator h and traffic flux indicator f. The hop count indicator h indicates the number of hop counts from the node to the root along the data collection tree and the traffic indicator f indicates the cumulated traffic flux along the path from the current node to the root.

Consider node o receives the path indicators (h_1, f_1) and (h_2, f_2) from its neighbor nodes i and j separately. Node o integrates both indicators to select the path. In this example, the estimated values for the two paths are given by

$$V_1 = (1-\alpha)\cdot h_1 + \alpha\left(\frac{h_1+h_2}{f_1+f_2}\right)\cdot f_1,$$
$$V_2 = (1-\alpha)\cdot h_2 + \alpha\left(\frac{h_1+h_2}{f_1+f_2}\right)\cdot f_2,$$

where the factor $(h_1+h_2)/(f_1+f_2)$ is used to normalize the hop count indicator with the traffic flux indicator. Thus, when there are N such path choices, the value for each path is estimated as

$$V_k = (1-\alpha)\cdot h_k + \alpha\left(\frac{\sum_{i=1}^{N} h_i}{\sum_{i=1}^{N} f_i}\right)\cdot f_k. \quad (5.1)$$

In such a way, the two indicators of hop count and traffic load are considered and weighted with a factor $\alpha \in [0,1]$, which is adjustable according to the application needs, i.e., a larger α when smoother traffic flux distribution is required and a smaller α when shorter data delivery path is required. After the path is determined, the current node o updates and passed downstream the two indicators $h_o = h_k + 1$ and $f_o = f_k + T_o$, where T_o is the traffic amount happened on node o.

As a result, when we integrate the traffic indicator into consideration we do create smoother traffic flux distribution over the field, shifting from the one that can be easily captured by the network flux model. Sampling on such a traffic distribution is, thus, unlikely to accurately calculate mobile users' locations, and the location privacy is preserved at some degree.

5.2 Reshaping Traffic Patterns

As the mobile users are moving within the network, we may not necessarily deliver the data to their instant locations. Instead, if we know or we can predict their moving trajectories, we can let sensor nodes deliver their data to ahead of the mobile users. In particular, we build data collection trees rooted at a set of discrete sensor nodes along the moving trajectory of each mobile user. We call them *hotspots*. Data are first aggregated to those *hotspots* and then the mobile user fetches the data from the *hotspots* when he moves along the trajectory. In such a way, we will average the data delivered to each mobile user along the moving trajectory and, thus, reshape the traffic pattern within the network.

When we build the data collection trees rooted at the *hotspots*, we consider both hop count indicators as well as traffic flux indicators, the same as what we describe in Section 5.1. The only difference is that the hop count indicator h is no longer associated with one root. Instead, the indicator h at each node records the hop count to the closest *hotspot*. In Appendix F, available in the online supplemental material, we present the algorithm executed at one mobile user. As a result, for each mobile user several data collection trees rooted at different *hotspots* are built and the data from an arbitrary sensor node are guaranteed to deliver to one of those *hotspots*. The traffic pattern can be further reshaped and averaged along the moving trajectories of mobile users.

In Appendix G, available in the online supplemental material, we give a further discussion on the effectiveness of the proposed two countermeasures.

6 Evaluations

We do extensive simulations to validate the effectiveness of the discussed attack and countermeasure approaches. We evaluate the accuracy of locating static users inside the network with NLS parameter fitting and tracking mobile users with Sequential Monte Carlo Estimation. We demonstrate the results with various inputs and examine the performance of the approach under different conditions.

In addition, we launch a trace driven experiment with the movement logs of mobile users in Dartmouth Campus data set [7] and evaluate the effectiveness of traffic redistributing and reshaping techniques in protecting the location privacy of mobile users. The detailed evaluation results can be found in Appendix H, available in the online supplemental material.

6.1 Instant Localization

To demonstrate the basic localization framework that we provide with the fingerprint information of network flux, we simulate a sensor network with 900 nodes on a 30 by 30 rectangular field. The sensor nodes are distributed over the field in perturbed grids [4]. The communication radius for each node is set to be 2.4, resulting in an average degree of 18. We simulate internal users within the field, collecting sensory data from the network. The traffic stretch of each user is randomly selected from 1 to 3. As described in Section 4.1, by doing the NLS fitting on the traffic flux over the network, we are able to approximate the locations of all internal users that are collecting data from the network.

In Fig. 3, we evaluate the localization accuracy of our NLS fitting-based approach with varied settings. We vary the percentage of sensor nodes that provide us flux samplings, testing the effectiveness of this approach with sparse inputs. For each percentage level, we randomly select the percentage of sensor nodes from the network and use their flux reports to calculate the locations of users.

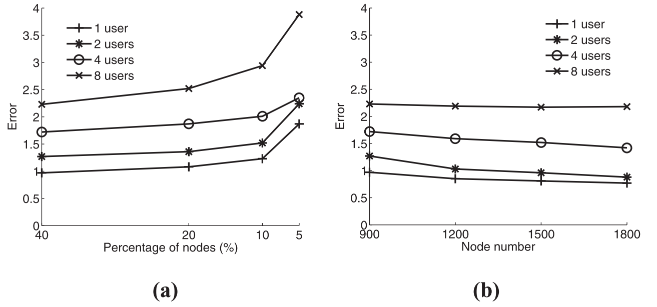

Fig. 3. Localization accuracy. (a) Against percentage of sampling nodes. (b) Against network density.

In Fig. 3a, we show how the localization accuracy varies with the percentage of sensor nodes we use. Consistent with our intuition, as the percentage of sampling nodes drops, the localization error increases. Nevertheless, the results prove that our approach is robust with sparse inputs. The localization error keeps low even when we only use the reports from 10 percent nodes. A dramatic increase of error happens when we further lower the usage of node samplings to 5 percent. The localization accuracy does not improve much when more than 40 percent nodes are sampled and on the other hand the localization accuracy dramatically decreases and becomes even unacceptable when the sampling nodes decrease below 5 percent . The number of simultaneous internal users affects the localization error. When we employ 10 percent nodes, our approach achieves localization error of 1.23 for one user, 1.52 for two users, 2.01 for four users and much increased 2.94 for eight users. We then vary the number of nodes deployed in the field from 900 to 1,800, resulting in different network densities. For this set of simulation, the node reports we use is fixed at 90. As Fig. 3b depicts, when the number of sampling nodes is fixed the localization error decreases as the network density rises. That is probably because in a denser network the proposed network flux model approximates the real network traffic more accurately, as we previously discussed in Section 3. The impact of network density, however, is fairly limited. The localization error does not significantly change with the network density.

6.2 Tracking Mobile Users

In this simulation, we let mobile users move within the field and track their moving trajectories by our Sequential Monte Carlo Estimation-based approach. The basic settings are the same as previous ones. In the Monte Carlo sampling process, we select $N = 1{,}000$ random samples every time and keep the top $M = 10$ samples as the updated representatives for the location of each user. At this stage, we assume that all mobile users simultaneously collect data with the same time interval, so we are able to test how accurate our approach will work with the complex traffic pattern assembled with multiple users. The maximum moving speed of each user is restricted below 5 per detection interval Δt, resulting in a resampling area of radius 5 each round.

We test the accuracy of our approach with different percentage of flux samplings and against different network densities. We measure the error of the location estimation of each user in the final round and depict the results in Fig. 4. We vary the number of mobile users from one to eight. As shown in Fig. 4a, the tracking accuracy does not vary much until the percentage of sampling nodes drops below 5 percent, which is consistent with what we observe in the localization scenario. Although a large number of flux samplings help to provide high accuracy, using only 10 percent nodes still provides us acceptable accuracy. In Fig. 4b, we depict how the tracking error varies with the network densities. The number of nodes deployed in the field is varied from 900 to 1,800 and the node reports we use is fixed at 90. Similar with the situation in the localization scenario, the network density does not significantly affect the tracking accuracy, although a denser network provides more accurate approximation with the network flux model.

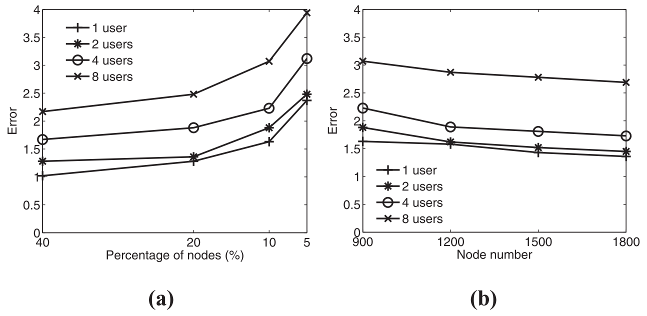

Fig. 4. Tracking accuracy. (a) Against percentage of sampling nodes. (b) Against network density.

7 Conclusions and Future Work

In this paper, we demonstrate that the mobile users within a sensor network take the risk of leaking their location privacy. The network flux provides external malicious entities fingerprint information about the mobile users inside the network. We propose a flux model that approximates the network flux within the network. We demonstrate that through passively sniffing a small set of nodes in the network, the external adversary can easily locate the mobile users and track their movement. This study reveals the potential threat in protecting the location privacy of mobile users from malicious entities. We then explore a set of traffic redistribution and reshaping methods and propose the countermeasures against such malicious attacks. The trace driven experiments suggest that the proposed approach effectively protect the location privacy from malicious attacks. A particular assumption we make throughout this paper is that the malicious sniffing spots are evenly distributed across the entire field, which might be a constraint to the attackers. We plan to investigate this problem and study how uneven sniffing over the field will work in future works.

Acknowledgments

This work is supported by NTU SUG M4080103.020 and NAP M4080738.020 in Nanyang Technological University, Singapore. A preliminary work has been published in IEEE ICDCS '10 [10].

References

[1] J. Al-Muhtadi, R. Campbell, A. Kapadia, M.D. Mickunas, and S. Yi, "Routing through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments," *Proc. IEEE 22nd Int'l Conf. Distributed Computing Systems (ICDCS),* 2002.

[2] P. Bahl and V.N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," *Proc. IEEE INFOCOM,* 2000.

[3] A.R. Beresford and F. Stajano, "Location Privacy in Pervasive Computing," *IEEE Pervasive Computing,* vol. 2, no. 1, pp. 46-55, Jan.-Mar. 2003.

[4] J. Bruck, J. Gao, and A.A. Jiang, "MAP: Medial Axis Based Geometric Routing in Sensor Network," *Proc. ACM MobiCom,* 2005.

[5] J. Deng, R. Han, and S. Mishra, "Countermeasures against Traffic Analysis Attacks in Wireless Sensor Networks," *Proc. First Int'l Conf. Security and Privacy for Emerging Areas in Comm. Networks (SECURECOMM),* 2005.

[6] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer,* vol. 34, no. 8, pp. 57-66, Aug. 2001.

[7] D. Kotz, T. Henderson, and I. Abyzov, "Trace Set Dartmouth/Campus/Movement (v1.3)," 2005.

[8] B. Kusy, A. Ledeczi, and X. Koutsoukos, "Tracking Mobile Nodes Using RF Doppler Shits," *Proc. ACM Fifth Int'l Conf. Embedded Networked Sensor Systems (SenSys),* 2007.

[9] B. Kusy, H. Lee, M. Wicke, N. Milosavljevic, and L. Guibas, "Perdictive QoS Routing to Mobile Sinks in Wireless Sensor Networks," *Proc. Information Processing in Sensor Networks (IPSN),* 2009.

[10] M. Li, X. Jiang, and L. Guibas, "Fingerprinting Mobile User Positions in Sensor Networks," *Proc. IEEE 30th Int'l Conf. Distributed Computing Systems (ICDCS),* 2010.

[11] M. Li, Y. Liu, J. Wang, and Z. Yang, "Sensor Network Navigation without Locations," *Proc. IEEE INFOCOM,* 2009.

[12] H. Lin, M. Lu, N. Milosavljevic, J. Gao, and L.J. Guibas, "Composable Information Gradients in Wireless Sensor Networks," *Proc. Seventh Int'l Conference Information Processing in Sensor Networks (IPSN),* 2008.

[13] S. Madden, M.J. Franklin, and J.M. Hellerstein, "TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks," *Proc. Fifth Symp. Operating Systems Design and Implementation (OSDI),* 2002.

[14] K. Madsen, H. Nielsen, and O. Tingleff, "Methods for Nonlinear Least Squares Problems," technical report, 2004.

[15] L.M. Ni, Y. Liu, Y.C. Lau, and A. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," *ACM Wireless Networks,* vol. 10, pp. 701-710, Nov. 2004.

[16] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall, "802.11 User Fringerprinting," *Proc. ACM MobiCom,* 2007.

[17] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," *Proc. ACM MobiCom,* 2000.

[18] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System," *ACM Trans. Information Systems (TOIS),* vol. 10, pp. 91-102, 1992.

Mo Li received the BS degree in the Department of Computer Science and Technology from Tsinghua University, China, in 2004 and the PhD degree in the Department of Computer Science and Engineering from Hong Kong University of Science and Technology. Currently, he is working as an assistant professor in the School of Computer Engineering of Nanyang Technological University of Singapore. He won ACM Hong Kong Chapter Professor Francis Chin Research Award in 2009 and Hong Kong ICT Award Best Innovation and Research Grand Award in 2007. His research interests include sensor networking, pervasive computing, mobile and wireless computing, etc. He is a member of the IEEE and ACM.

Xiaoye (Jonathan) Jiang is working toward the PhD degree at Stanford University in the Computational Mathematics Department. His current research focuses on algebraic methods in machine learning and data mining with applications in multiobject tracking, ranking and social network studies. He is a student member of the IEEE.

Leonidas J. Guibas received the PhD degree from Stanford in 1976, under the supervision of Donald Knuth. His main subsequent employers were Xerox PARC, MIT, and DEC/SRC. He has been at Stanford since 1984 as a professor of computer science. He has produced several PhD students who are well known in computational geometry, such as John Hershberger, Jack Snoeyink, and Jorge Stolfi, or in computer graphics, such as David Salesin and Eric Veach. At Stanford he has developed new courses in algorithms and data structures, geometric modeling, geometric algorithms, and sensor networks. He is an ACM fellow and winner of the ACM/AAAI Allen Newell Award. He is a fellow of the IEEE.