Hiding Trajectory on the Fly

Xinyu Jin[1], Niki Pissinou[1], Cody Chesneau[2], Sitthapon Pumpichet[1], Deng Pan[1]
[1]Florida International University
{First name.Last name}@fiu.edu
[2]Wofford College
chesneauce@email.wofford.edu

***Abstract*—The rapid development in micro-computing has allowed implementations of complex mobile Wireless Sensor Networks (mWSNs). Privacy invasion is becoming an indispensable issue along with the increasing range of applications of mWSNs. Private trajectory information not only indicates the movements of mobile sensors, but also reveals personal preferences and habits of users. In this paper, we propose the distributed Basic Trajectory Privacy (BTPriv) and Secondary Trajectory Privacy (STPriv) preservation algorithms to hide trajectory of data source nodes online. We set up various simulation environments for different applications. The effectiveness of our proposed algorithms is evaluated by the software implementation in simulation experiments.**

Keywords- trajectory privacy; mobile sensor network; online algorithm

I. INTRODUCTION

Wireless Sensor Networks (WSNs) have rapidly been developed in a wide range of applications. Mobile WSNs (mWSNs) relax the restrictions on node density and static node locations [10] comparing to stationary WSNs. It makes the implementation of applications much more efficient, effective, and inexpensive. On the other hand, trajectory privacy is a critical concern for these applications. When sensors are carried by human beings and vehicles, trajectory information of the mobile sensor reveals the private trajectory of the user to unauthorized entities, which may seriously threaten user's personal safety. Moreover, trajectory information can reveal personal preferences and habits, which can be used for consumer profiling by such as insurance companies.

The aim of this work is to develop a mechanism which hides the trajectory of data source nodes, denoted as target nodes, on the fly with considering nodes mobility in WSNs. In this particular paper, "on the fly" is defined as a node hiding its trajectory while undergoing data transmissions. We propose the unique privacy-aware routing phase, where each node selects the next-hop node according to dynamic trajectory distance to hide its trajectory. To the best of our knowledge, this is one of the first works, if it is not the very first, to provide distributed and online trajectory privacy preservation mechanisms in mWSNs. We summarize our contributions in the following.

- Define the passive trajectory privacy invasion model in mWSNs.
- Develop the one-time pad virtual name to hide trajectory of target nodes without the third party.
- Create the unique privacy-aware routing phase for hiding trajectory of target nodes on the fly.

The remainder is structured as follows. Related works will be reviewed in section 2. The proposed methods will be presented in section 3 and 4, followed by simulation results. Finally, we will discuss limitations and future work.

This work was partially supported by NSF grants CNS-0843385, CNS-0851733 and Department of Homeland Security.

II. RELATED WORK

This project deals with trajectory privacy of mobile source nodes in sensor networks in the online manner. Works focusing on offline applications will not be discussed here.

One technique to hide location privacy of source nodes in WSNs is random walk [13], [23]. The message is routed in a random or directed random fashion before it is flooded or routed to the sink. Researchers also proposed to randomly select intermediate nodes with a minimum distance from source nodes to achieve the randomness of routing [14]. Another technique is adding noise, including adding dummy messages [13], [20], [26], and simulate fake source nodes [16]. This technique gives excess power consumption for sensor network. The third technique is using cryptographic techniques to encrypt users' identities with lowered overhead [18]. With some background knowledge, the attacker could still crack the encrypted identity by linking the background with the user. The method to protect temporal privacy of message generating is to locally buffer data for a random time period at intermediate sensors [12]. Transmission delays in such networks are significant and difficult to control. Moreover, the critical limitation for all above techniques to be used in mWSNs is that they are restricted by fixed locations of the network components. With consideration on users' mobility, *k*-anonymity is commonly used in many previous works [19], especially for Location Based Service (LBS). Researchers proposed a framework based on *k*-anonymity [3]. Building upon this framework, researchers studied several methods to compute different Clocking Regions (CKs), including pyramid data structure based CK [17], [22], Hilbert curve based CK [7], [4], temporal-spatial box [5], and circle CK [25]. The personalized trajectory *k*-anonymity in network-confined environment was provided in [1]. *K*-anonymity is vulnerable to background knowledge attack, query sampling attack and query tracking attack [15]. To address these attacks, researchers proposed the concepts of *l*-diversity [15], reciprocity [11], [2], and memorization property [2]. The difficulty to apply *k*-anonymity in sensor networks is how to design the anonymizer. It is not practical for resource-constrained mobile sensor networks with highly dynamic architectures. Distributed algorithms are more desirable. The design in [6] removed anonymizers. However, how to secure the index map was not considered. Moreover, all above anonymity-based algorithms are vulnerable to distributed eavesdropping attacks modeled in the following section.

III. PRELIMINARIES

A. System Model

The considered network is the mobile sensor network with stationary backbone infrastructures for realistic applications. The Base Station (BS) is the sink to receive data. Multiple Access Points (APs) are interconnected through a backbone network and connected to the BS. The network is partitioned

into N distinct subareas $SUB = \{Sub_1, Sub_2, \dots, Sub_N\}$.The AP is located at the center of a subarea Sub_i and serves nearby nodes within the entire subarea, named correspondingly AP_i. Nodes send data through one or multiple hops to reach one of the APs. Then APs forwards data to the BS. We assume the physical location distance is equivalent to routing path distance. Mobile nodes are homogenous sensors with diverse trajectory patterns and moving velocities. Nodes send data to the nearby APs; include both sensing data and trajectory for system localization. The mobility is provided by carriers, named users, such as human bodies and cellular phones. We consider $U = \{u_1, u_2, \dots, u_k\}$ a set of K nodes moving within SUB during a time period $T = \{t_1, t_2, \dots, t_T\}$. The accurate trajectory of node u is T_{ru}, defined in the next section. The real trajectory set of all K nodes is $R = \{T_{r1}, T_{r2}, \dots, T_{rk}\}$. The approximate trajectory set of node u is $r_u = \{Sub_{ut_1}, Sub_{ut_2}, \dots, Sub_{ut_T}\}$ during T.

B. Adversarial Model

We developed this project with consideration of the passive attack. Inspired by the framework in [21], we characterize the adversary by his knowledge and attack.

Knowledge – The adversary has adequate computation capability, energy and memory for data storage. The adversary does not have the knowledge of encrypted sensor data. However, data packets headers are usually left unencrypted for routing purposes where the source identity is revealed. Moreover, the adversary could easily obtain some public information of users, such as working and home addresses. Therefore, the adversary could even crack encrypted identities. The adversary is also assumed to know some system parameters, include the network partition SUB and user set U. We define the whole knowledge of the adversary AK.

Attack – The adversary behaves in honest-but-curious [9] model. He deploys multiple stationary nodes to eavesdrop wireless communications in the network. Preferred locations of these eavesdroppers are nearby APs, where sensor data streams are aggregated. In shortest-path oriented routing, the adversary can deduce that the source node is within a certain subarea when he "hears" the corresponding packets. The adversary monitors traffics over the entire network. In other words, this is a global adversary. The adversary does not launch active attacks. Node compromise attack is excluded. The objective of the adversary is to find out the whole trajectory of nodes – tracking attack; or to localize a node at a given time instant – localization attack. This node is the target node. We consider data source nodes as target nodes. The trajectory set of all K users observed by the adversary is $R' = \{T'_{r1}, T'_{r2} \dots, T'_{rk}\}$. The approximate trajectory set of node u in time period T observed by the adversary is $r_u' = \{Sub'_{ut_1'}, Sub'_{ut_2'} \dots, Sub'_{ut_T'}\}$. All observations of the adversary is denoted as O.

Fig. 1 is an illustration of trajectory privacy invasion. A user carries a sensor moving in the resident area. Eavesdroppers are shown as red dots. The black curve is the real trajectory of the user, while the red straight line is estimations by the adversary. The comparison is shown in Table 1. The adversary may even perceive that the user visited the hospital and the school, and passed by the park and the bank without a stop, by comparing the amount of packets heard at different APs.

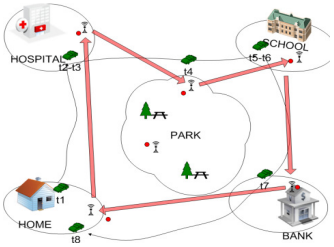

Figure 1. Trajectory privacy invasion

TABLE I. TRAJECTORY OBTAINED BY EAVESDROPPERS

T	r_u	r_u'
t1	HOME	HOME
t2-t3	HOSPITAL	HOSPITAL
t4	TRAVEL	PARK
t5-t6	SCHOOL	SCHOOL
t7	TRAVEL	BANK
t8	HOME	HOME

C. Objective and Evaluations

With the knowledge AK and observation O, the adversary is trying to reconstruct R and r correctly. If the adversary is able to localize users to specific subareas in time period T, he will also successfully track users. Our objective is to prevent the adversary from reconstructing r for each user. The evaluations metrics are developed based on adversarial evaluation metrics in [21]. We define our evaluation metrics in the following.

Let $x_{ut}', u \in U$ and $t \in T$, be the estimate of the subarea that node u was located in at time t. For all target nodes during T, all estimates based on the observation O comply with the probability distribution $P(X = x_{ut}'|O)$, $x_{ut}' \in SUB$. The real subarea that u was located in is denoted as $x_{ut} \in SUB$. The *incorrectness* E, which is the adversary's expected estimation error, can be quantified using the expected distance between x_{ut} and x_{ut}' with probability distribution $P(X|O)$. For example, this distance can be the distance between AP_{ut} and AP_{ut}'. Assuming the coordinates of these APs in 2D space are (a_{ut}, b_{ut}) and (a_{ut}', b_{ut}'), respectively, the incorrectness can be computed as follows:

$$E = \sum_{x_{ut}'} P(X = x_{ut}'|O) \sqrt{(a_{ut} - a_{ut}')^2 + (b_{ut} - b_{ut}')^2}. \quad (1)$$

The incorrectness E of the adversary is the metric that evaluates the trajectory privacy of the system. The higher E is, the higher trajectory privacy is.

Recall that the adversary in our model is a global adversary. It is possible that he could estimate trajectories by analyzing traffic distribution on the network. For instance, if packets from target nodes are "heard" by certain eavesdroppers with obvious higher probability, the adversary will be able to learn that the real trajectories have special connections with the these subareas. The estimated trajectories could be constrained by the special connections. We use entropy of the distribution $P(X = x_{ut}'|O)$, $x_{ut}' \in SUB$ to evaluate the performance of our algorithms against the global adversary which is computed by:

$$H = -\sum_{x_{ut}'} P(X = x_{ut}'|O) \log P(X = x_{ut}'|O). \quad (2)$$

The higher the entropy is, the more uniform the distribution is. It gives lower probability to find out the special connections.

IV. PROPOSED METHODS

A. Problem Definition

We first present a definition of trajectory, which is often used in Moving Objective Database (MOD) literature [8]. For simplicity, we assume that nodes are moving in 2D space.

Definition 1: A trajectory T_r of a moving node is a polyline in the three-dimensional space, where two dimensions refer to space and the third dimension to time. It is represented as a sequence of points $< (x_1, y_1, t_1), \dots, (x_n, y_n, t_n) >$ *with* $t_1 < \cdots < t_n$.

Each point (x_i, y_i, t_i) in the sequence represents the 2D location (x_i, y_i) of the node, at time t_i. Observe that the trajectory above is defined in MOD, where all information is collected into the offline centralized database for trajectory analysis. In order to hide the node's trajectory on the fly in mWSNs, we need to analyze the dynamic trajectory distance between two nodes on the network.

The dynamic trajectory distance represents the irrelevance of two trajectories in terms of 2D location and velocity at a specific time. At time t, given the 2D location (x_{it}, y_{it}) and velocity vector $(\overrightarrow{i_t})$ of node i , and the 2D location (x_{jt}, y_{jt}) and velocity vector $(\overrightarrow{j_t})$ of node j, the location distance between i and j

$$\mathrm{D_l(i,j,t)} = \frac{\sqrt{(\mathrm{x_{it}}-\mathrm{x_{jt}})^2+(\mathrm{y_{it}}-\mathrm{y_{jt}})^2}}{\mathrm{r_{max}}} . \quad (3)$$

r_{max} is the maximum transmission distance of a node in the network. The velocity direction distance is derived from the cosine similarity of two vectors, which is

$$\mathrm{D_v(i,j,t)} = 1 - \mathrm{Sim}(\overrightarrow{\mathrm{i_t}}, \overrightarrow{\mathrm{j_t}}) = 1 - \frac{\overrightarrow{\mathrm{i_t}}\cdot\overrightarrow{\mathrm{j_t}}}{\|\overrightarrow{\mathrm{i_t}}\|\|\overrightarrow{\mathrm{j_t}}\|}. \quad (4)$$

If $\|\overrightarrow{i_t}\| \cdot \|\overrightarrow{j_t}\| = 0$, $D_v(i,j,t) = 0$. The unified speed of node j

$$\mathrm{S(j,t)} = \frac{\|\overrightarrow{\mathrm{j_t}}\|}{\mathrm{S_{max}}}. \quad (5)$$

S_{max} is the possible maximum speed of sensor nodes. Finally, we have the following:

Definition 2: The Dynamic Trajectory Distance represents the irrelevance of two trajectories in terms of 2D location and velocity at a specific time. Given $D_l(i,j,t)$ and $D_v(i,j,t)$ between the target node i and its neighbor node j, and $S(j,t)$ of the neighbor j, the dynamic trajectory distance

$$\mathrm{D_T} = \mathrm{w_1 D_l(i,j,t)} + \mathrm{w_2 D_v(i,j,t)} + \mathrm{w_3 S(j,t)}. \quad (6)$$

$W(w_1, w_2, w_3)$ is a weighted vector and $w_1 + w_2 + w_3 = 1$.

B. Privacy-aware Routing

From Fig.1, we observe the following: 1. Without breaking the network security, adversaries could estimate the target node's trajectory through simply eavesdropping messages en route at low cost; 2. Adversaries could obtain trajectory information during message transmission on the network without accessing the BS; 3. Conventional shortest-path oriented routing protocol (for simplicity, we will use conventional routing protocol in the rest of the paper) creates the possibility for adversaries to deduce trajectory information. We propose to use Trajectory Privacy-aware routing (TPriv) to hide the target node's trajectory on the fly.

TPriv has two different phases for message routing: conventional routing phase and privacy-aware routing phase. For the message requesting high priority in transmission delay, the conventional routing phase is only needed for data transmissions. Packets are routed according to the shortest-path routing algorithm. In mobile sensor networks, due to the high dynamic network topology, on-demand oriented routing algorithms are recommended. For regular data, which are collected by the BS periodically, the privacy-aware routing phase is required, followed by the conventional routing phase. The privacy-aware routing phase is detailed as follows.

The objective of this phase is intuitive and effective: Through forwarding the message to nonlocal APs, it prevents trajectory privacy invasion by eavesdropping attack at network APs. To achieve this objective, the target node needs to select the next hop under the principle that among all the neighboring nodes the candidate has the highest probability to forward the packet to a nonlocal AP. To meet this principle, two difficulties need to be overcome as an online distributed design: The target node does not have the trajectory information of its neighboring nodes as background to select the next hop; assuming the first problem is overcome, the trajectory privacy of neighboring nodes can be breached by the target node. TPriv overcomes both these difficulties and meets the design principle. The target node collects limited information from its neighbors before data transmissions in the query-and-reply fashion. In order to prevent the target node from breaching other nodes' trajectory privacy, one-time pad Virtual Identity (VID) is deployed during trajectory query-and-reply. Then each neighbor is evaluated in terms of its dynamic trajectory distance to the target node. Finally, sensed data are routed to a nonlocal AP through the selected next hop.

1) Dynamic trajectory information collection

During this stage, two types of VID will be used: Query VID (QVID) and Reply VID (RVID). Both are one-time pads that generated by the node itself. Each VID is valid for a certain period of time. The life time for QVID $LT_Q = \Delta T$, while $LT_R = 2\Delta T$ for RVID, where ΔT is the time difference between two transmissions. LT_Q is designed to prevent the target node from sending continuous queries during one message transmission interval which may lead to a waste in bandwidth or even network congestion. LT_R is designed to prevent the target node from selecting one neighboring node as the next hop consecutively. The VID is not used for data transmissions. There is no need to set up the third party [3] for virtual name mapping between nodes and the BS.

Before the target node transmits data, it broadcasts a dummy query under its QVID. Each one-hop neighbor replies it under its RVID. The reply contains the dynamic trajectory information. During time ΔT, any duplicate query message with the same QVID will be ignored by neighbors. After the target node receives the reply, it temporarily stores the dynamic trajectory information except the entry that has the same RVID as the selected next hop during the last data transmission.

2) D_T computation and the next-hop node selection

Next-hop nodes could be more than one, which may provide better performance in terms of privacy. However, there is a tradeoff between privacy and power consumption. So far, we have considered one next-hop node only. The target node computes D_T to each neighbor according to (6), respectively. Since D_T represents the irrelevance of two trajectories in terms of the 2D location and velocity at a specific time, the neighbor, which has the largest D_T to the target node will be selected as the next-hop node at the time of each data transmission. Assuming D_T computation time is negligible, the candidate remains the same location and velocity.

After the selected next hop receives the message, it resets the hop count entry in the packet header to be 0 and starts the conventional routing phase. With the implementation of TPriv, the target node routes each regular message to a nonlocal AP with a certain probability. It misleads the passive privacy adversary at each time of data transmission. However, the probability of reaching nonlocal APs with one-hop privacy-

aware routing phase is undesirable in large-scale networks. Moreover, the frequency of each nonlocal AP to be en route may disclose the trajectory of the target node to global adversaries. Therefore, we propose the Secondary TPriv (STPriv) to resolve these problems. The above method is referred as Basic TPriv (BTPriv).

C. Secondary TPriv

STPriv improves BTPriv in the following aspects:

1) M-hop privacy-aware routing phase

The basic TPriv only provides a very limited range for finding nonlocal APs. Therefore, we invent the *m*-hop privacy-aware routing phase, which requires the following *m* next hops will be selected in a privacy-aware fashion. *M* is a predefined parameter, depending on the network deployment. Here we take *m* equal to 2 as an example. Upon receiving packets from the target node, the intermediate node takes similar procedures as the target node, except for message filtering and dynamic trajectory distance computing.

a) Message filtering

To avoid a routing loop, in the case that any intermediate node routes the message back or closer to the previous-hop node, the intermediate node broadcasts the query message with the same QVID as the previous-hop node. Any neighbor who receives the duplicate query message consecutively ignores this query. Therefore, the next-hop selection is constrained to the neighbors at least two hops away from the target node.

b) Dynamic trajectory distance (D_T') computation

For forwarding packets further, the second hop candidate with greater D_l, higher speed and smaller D_v, is preferred. This is because the current intermediate node already has very different velocity direction from the target node after the basic privacy-aware routing phase. D_T' is computed as follows:

$$\mathrm{D_T'} = \mathrm{w_1 D_l(i,j,t)} + \mathrm{w_2}(1 - \mathrm{D_v(i,j,t)}) + \mathrm{w_3 S(j,t)} \qquad (7)$$

The greater *M* does not indicate better performance. *M* dominates the balance between privacy and transmission delays. It depends on the network topology and application.

2) Randomized probability for TPriv

BTPriv requires that all regular messages are routed in the privacy-aware phase first. Therefore, the target node which always moves among certain subareas will hardly transmit data through the corresponding local APs. For the global adversary it will be obvious that the target node is moving within certain subareas. To address this issue, STPriv allows the target node to choose the conventional routing phase directly with a certain probability *p*. The value depends on the network scale.

V. Simulation Results

The proposed algorithms were implemented using Python and Matlab on Windows XP on a 2.13 GHz Intel Core 2 CPU equipped with 2GB of main memory. We consider a square area divided into 40X40 m^2 subareas to represent institution network fields. The network scale varies among containing 4, 9 and 16 subareas. Nodes are moving in the manner of random waypoint. Nodes are categorized into three types according to their trajectory patterns: restricted nodes (moving within one subarea), repeating node (repeating certain trajectory in some subareas), and traversing nodes (randomly traversing the entire network). Nodes are deployed by given random locations in the network grid and the speed between 0.2-22 mph. We also use a 200X200 m^2 area within DHDN/3-degree Gauss-Kruger zone 2 (EPSG code: 31466) to represent part of city-wise network areas. It contains 9 subareas of interests. The trajectory data of nodes are generated by using the Random Street model of BonnMotion [24]. The weighted vector, $W = [0.5, 0.2, 0.3]$. The maximum transmission range of each node is set to be 20 m, referenced to the average transmission range for reliable connection in our real experiments on MEMSIC sensors equipped with MTS420 boards. We conducted 20 independent rounds of simulations for each set of parameters. In one round, each data source node transmits 100 packets. Since we focus on the application and network layers, some nodes in the network are set to be dummy nodes, which only forward packets, to avoid channel collision.

Recall the evaluation metrics we defined in section 3, we firstly evaluate the performance of our design by computing attackers' incorrectness value *E* according to (1). The method to define the distance between $\mathrm{x_{ut}}$ and $\mathrm{x_{ut}}'$ could vary in different applications and spaces. In order to have straight forward evaluation results, here we define the distance between x_{ut} and x_{ut}' is 0 if and only if $x_{ut} = x_{ut}'$. Otherwise, the distance is 1. Then, we have $\mathrm{E} = 1 - \mathrm{P}(\mathrm{x_{ut}}|\mathrm{O})$.

Fig.2 is the box-and-whisker plot by implementing BTPriv. The data were collected by considering all the repeating and traversing as target nodes. The mean of *E* increases along with the average node density and the network scale. BTPriv has less time and power consumption comparing to STPriv in terms of number of hops for privacy-aware routing phase. After the node density reaches to a certain limit, *E* slightly increases. With the improvements from *M*-hop STPriv, the value of E dramatically increases. We simulated 2 to 5 hops of STPriv. The result (mean $\pm$ standard deviation) is shown in Fig.3 (we use line plot to represent discrete data for better illustration). The node density is 1 node/100 m^2 (as well as in Fig. 4 and Fig. 6). It is observed that after *M* is greater than 3, the performance of STPriv keeps stable. We have defined *p* close to 1/2*N* for choosing conventional routing algorithm. Even though, the stable point still reaches 81% in 16-subarea network. With 5-hop STPriv implementation, *E* is as high as 92.6%. In other words, the adversary fails in locating target nodes for 92.6% of the time. Fig. 5 and Fig. 6 show the performance of STPriv in the city-wise network. The average node speed used in Fig. 5 is 20 mph (as well as in Fig. 4(b)). Similar to small-scale networks, STPriv performs better when *M* increases. STPriv also has stable performance when node density and average node speed vary.

Next, we evaluate the performance of preventing global adversaries by computing the entropy *H* according to (2). The ideal distribution $P(X|O)$ is uniform distribution. For example, in a 4-sub network, $P(X = x_{ut}'|O) = 0.25 \; \forall \; x_{ut}' \in SUB$ and the normalized entropy $H = 1$. Fig. 4(a) shows the simulation results in small-scale networks. For repeating nodes and traversing nodes (Rp & T), BTPriv offers comparable performance as the ideal distribution. *H* is as high as 0.999. In Fig. 4(b), STPriv also gives high *H* value with small *M* in city-wise networks. This is due to the randomness of nodes movements. However, more hops in the privacy-aware routing phase are necessary for hiding trajectory of restricted nodes (R), shown in Fig. 4(a). The corresponding normalized entropy is computed and also shown in Fig.4. With greater *M*, STPriv is also effective to prevent global adversaries in extreme cases.

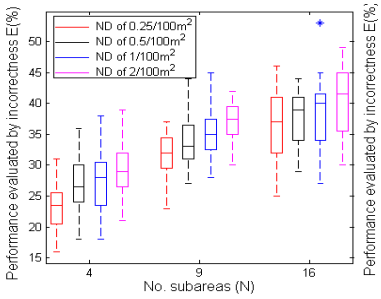

Figure. 2 Performance of BTPriv in different scales of networks

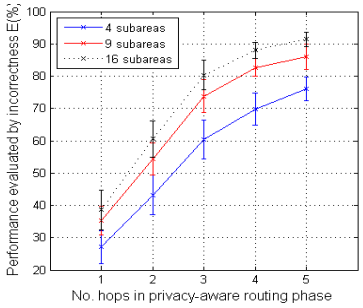

Figure. 3 Performance improvement of STPriv

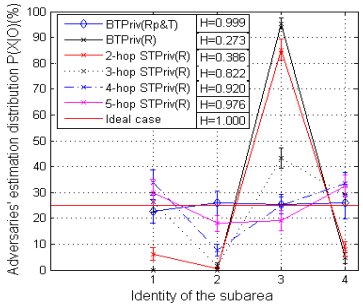

Figure. 4(a) Performance of preventing global adversaries in small-scale networks

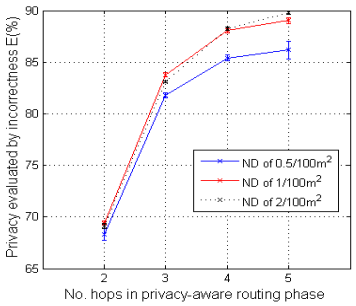

Figure. 5 Node density impact on STPriv in city-wise networks

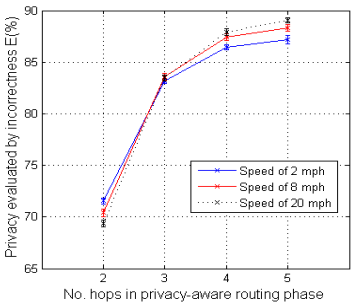

Figure. 6 Node speed impact on STPriv in city-wise networks

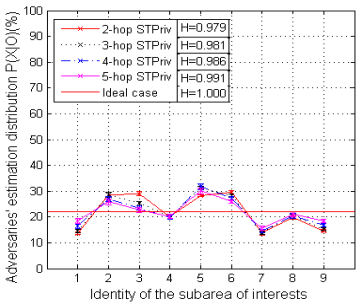

Figure. 4(b) Performance of preventing global adversaries in city-wise networks

VI. CONCOLUSION

In this paper, we defined the passive trajectory privacy invasion model in mWSNs and presented TPriv to hide the trajectory of data source nodes from privacy adversaries. We defined the proper evaluation metrics to evaluate the trajectory privacy performance of the above methods. One limitation of TPriv is the extra power consumption for query-and-reply during the privacy-aware routing phase. Although the development of both micro-computing and nanotechnology is resolving the power issue to a large extent, real applications are still lagging. Our future work is to improve this limitation and evaluate the network performance in terms of transmission delays and power consumption.

REFERENCE

[1] A. Gkoulalas-Divanis, V.S. Verykios, M.F. Mokbel, Identifying Unsafe Routes for Network-Based Trajectory Privacy. In Proc. SDM, 2009.

[2] C. Y. Chow and M. F. Mokbel. Enabling Private Continuous Queries for Revealed User Locations. In Proc. SSTD 2007.

[3] R. Cheng, Y. Zhang, E. Bertino and S. Prabhakar. Preserving User Location Privacy in Mobile Data Management Infrastructures. Workshop on Privacy Enhancing Technologies, 2006.

[4] M. L. Damiani, E. Bertino, and C. Silvestri. PROBE: an Obfuscation System for the Protection of Sensitive Location Information in LBS. CERIAS Technical Report, Purdue University, 2008.

[5] B. Gedik and L. Liu. Location Privacy in Mobile Systems: a Personalized Anonymization Model. In Proc. IEEE ICDCS 2005.

[6] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K. Tan. Private Queries in Location Based Services: Anonymizers are not necessary. In Proc. ACM SIGMOD 2008.

[7] G. Ghinita, P. Kalnis, and S. Skiadopoulos. MobiHide: A Mobiea Peer-to-Peer System for Anonymous Location-Based Queries. In Proc. SSTD 2007.

[8] R. H. Güting and M. Schneider. Moving Objects Databases. Morgan Kaufmann, 2005.

[9] O. Goldreich. The foundations of Cryptography – Volume 2. Cambridge University Press, 2004.

[10] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, and M. Geraczko. CarTel: A Distributed Mobile Sensor Computing System. In ACM Sensys 2006.

[11] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, Preventing Location-Based Identity Inference in Anonymous Spatial Queries. IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 12, pp 1719-1733, December 2007.

[12] P. Kamat, W. Xu, W. Trappe, and Y. Zhang. Temporal Privacy in Wireless Sensor Networks. In Proc. IEEE ICDCS 2007.

[13] P. Kamat, Y. Zhang, W. Trappe, C. Ozturk. Enhancing Source-Location Privacy in Sensor Network Routing. In Proc. IEEE ICDCS 2005.

[14] Y. Li and J. Ren. Source-Location Privacy through Dynamic Routing in Wireless Sensor Networks. In Proc. IEEE INFOCOM 2010.

[15] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: Privacy Beyond K-anonymity, In Proc. IEEE ICDE 2006.

[16] K. Mehta, D. Liu, and M, Wright. Location Privacy in Sensor Networks Against a Global Eavesdropper. In Proc. IEEE ICNP 2007.

[17] M. F. Mokbel, C. Y. Chow, and W. G. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In Proc. VLDB 2006.

[18] K. Pongaliur and L. Xiao. Maintaining source privacy under eavesdropping and node compromise attacks. In Proc. IEEE INFOCOM 2011.

[19] L. Sweeney. K- Anonymity: A Model for Protecting Privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, vol.10, no. 5, pp 557-570, 2002.

[20] M. Shao, Y. Yang, S. Zhu, G. Cao. Towards Statistically Strong Source Anonymity for Sensor Networks. In Proc. IEEE INFOCOM 2008.

[21] R. Shokri, G. Theodorakopoulos, J. L. Boudec, J. Hubaux. Quantifying Location Privacy. In IEEE Symposium on Security and Privacy, 2011.

[22] T. Xu and Y. Cai. Location Cloaking for Safety Protection of Ad Hoc Networks. In Proc. IEEE INFOCOM 2009.

[23] Y. Xi, L. Schwiebert, and W. Shi. Preserving Source Location Privacy in Monitoring-Based Wireless Sensor Networks. In Proc. IEEE IPDPS, 2006.

[24] BonnMotion. University of Bonn, Germany. Available: http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/.

[25] M. L. Yiu, C. S. Jensen, X. Huang and H. Lu, SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services. In Proc. IEEE ICDE 2008.

[26] Y. Yang, et al. Towards Event Source Unobservability with Minimum Network Traffic in Sensor Networks. In Proc. ACM WiSec 2008.