Catching All Pokémon: Virtual Reward Optimization With Tensor Voting Based Trajectory Privacy

Xinyue Zhang, *Student Member, IEEE*, Jingyi Wang, *Student Member, IEEE*, Yong Li, *Senior Member, IEEE*, Riku Jäntti, *Senior Member, IEEE*, Miao Pan, *Senior Member, IEEE*, and Zhu Han, *Fellow, IEEE*

***Abstract*—The widespread adoption of mobile devices with global positioning system enables location-based games (LBGs) to use real world maps, while locations and objectives in LBGs can make the progression, achievements, and virtual rewards feel more palpable and entertaining. However, allowing location sharing in LBGs gives dishonest parties opportunities to learn users' trajectories, which compromises the users' privacy. In this paper, we propose a novel scheme jointly maximizing LBG players' virtual rewards while preserving their trajectory privacy. Briefly, we first introduce a quantitative machine learning-based approach to model trajectory inference attacks via tensor voting. Then, to thwart this attack, we propose a tensor voting-based k-anonymous obfuscation strategy. Considering the trajectory privacy concerns and power constraint of hand-held mobile devices, we mathematically formulate the LBG players' virtual reward maximization optimization into the mixed integer problem and develop the feasible solutions. Simulation results and analysis show that the proposed scheme can effectively preserve LBG players' trajectory privacy against tensor voting based inference attacks while maximizing LBG players' virtual rewards.**

***Index Terms*—Virtual reward maximization, tensor voting, trajectory privacy, obfuscation, inference attacks.**

I. Introduction

RECENTLY, the rapid development of wireless communication, the exploding growth of smart devices and the universal use of global positioning system (GPS) have spurred the proliferation of the location-based services (LBSs). Based on users' location information, LBSs offer useful features from location-based discovery tools and smart search to games and exercise tracking (e.g., Foursquare, Yelp, Glympse, Detour, Gowalla, Shopkick, SCVNGR, etc.). LBSs help users keep up with friends, discover what is nearby, save money and unlock deals, play location based augmented reality games, keep healthy habits, etc. Actually, within the last few years, LBSs have penetrated into every corner of modern people's daily life, and brought us convenience. Among all LBSs, the location based game (LBG) is a rocket-soaring business. Taking one of the most popular LBGs, Pokémon Go [1], initially released in selected countries in July 2016, for example, it has been widely reported that Pokémon Go surpassed 1 billion dollars in revenue by February 2017 (just over six months) and has 65 million monthly active users. By using real world maps, LBGs make augmented reality more approachable and provide more palpable fun to the LBG players. Those extremely addicting LBGs (e.g., Pokémon Go, Ingress, Zombie Run, Resources Games, Parallel Kingdom, etc.) have made the players become the "snake" in Nokia Snake game, wandering around just to maximize the virtual reward.

Despite the crazes of LBGs, LBGs' prerequisites of players' location sharing raise serious privacy concern. Generally speaking, most LBGs require the player's hand-held device location, periodically report the location information to the LBG service provider, and the location data will be stored in the servers of the LBG provider. This implies that after accepting the terms to share their locations, the players have no control of deleting or modifying those data. Following the location reporting mechanism above, dishonest third-party LBG servers or eavesdropping attackers may have chances to know the user's reported locations, and leverage those locations to infer the trajectory of the user [2]. With the exposure of trajectory, the players not only lose their privacy but also are vulnerable to various attacks, even serious physical attacks. That is, the player's trajectory can be inferred in the digital world, so that his/her privacy will be invaded in the real world. A real life example happened in Missouri, July 2016 is that 11 Pokémon Go players have been ripped off because of playing this LBG, as thieves learned their trajectory and lured those victims to remote areas outside of St. Louis. Besides robbery, more serious crimes such as sexual assault, kidnapping, murder, assassination, etc. become possible targeting specific victims at selected locations.

To prevent the crime from happening again/before it happens, it is worthwhile to investigate how the malicious parties, either dishonest LBG providers or eavesdropping attackers, analyze the location data, and infer the players' trajectories. Based on the understanding of inference attacks, it is necessary to develop corresponding trajectory privacy preservation schemes for LBG

Manuscript received May 9, 2018; revised September 12, 2018; accepted November 1, 2018. Date of publication November 21, 2018; date of current version January 15, 2019. This work was supported in part by the U.S. National Science Foundation under Grants CNS-1343361, CNS-1350230 (CAREER), CNS-1646607, CNS-1702850, and CNS-1801925 and in part by the Academy of Finland project AIMHIS (decision no. 311760). The review of this paper was coordinated by Dr. A. Chatterjee. *(Corresponding author: Miao Pan.)*

X. Zhang, J. Wang, M. Pan, and Z. Han are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204 USA (e-mail: xzhang67@uh.edu; jwang86@uh.edu; mpan2@uh.edu; zhan2@uh.edu).

Y. Li is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: liyong07@tsinghua.edu.cn).

R. Jäntti is with the Department of Communications and Networking, Aalto University, Aalto FI-00076, Finland (e-mail: riku.jantti@aalto.fi).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2018.2882733

players. Meanwhile, it will be great for the LBG players to maximize their virtual rewards in the LBGs (e.g., catching all rare Pokémons nearby in Pokémon Go game) without any trajectory privacy leakage. Besides, there is a lack of quantitative approaches to analyze either inference attacks or the corresponding privacy preservation methods in previous studies.

To address those challenges above, in this paper, we introduce a novel trajectory inference attack model based on tensor voting theory [3], which can quantitatively model and analyze trajectory inference attacks. To thwart tensor voting based inference attacks, we propose a new trajectory privacy preserving approach. To satisfy LBG players, we further develop a virtual reward optimization scheme under trajectory privacy preservation and smart devices' energy consumption constraints. Our salient contributions are summarized as follows.

- Different from the existing trajectory inference attacks [4], we consider a novel trajectory inference attack model based on tensor voting theory. Using tensor voting based attacks, the adversary can infer LBG players' trajectories based on limited/partial information, e.g., just some location data without any timestamps. As tensor voting is robust and sensitive to Gaussian noise, the outlier locations are supposed to be filtered out after applying tensor voting. Thus, under tensor voting based inference attacks, those trajectory privacy preserving schemes with random noises are not applicable.
- Against the tensor voting based trajectory inference attacks, we propose a dummy-based k-anonymous trajectory privacy preserving scheme. In our scheme, we choose dummy locations from a location set which includes Pokémon locations, gyms in the game and other candidate locations, which is able to satisfy k-anonymous requirements. In order to make progress on the privacy quantification of the performance of the proposed trajectory privacy-preserving solutions, we quantify the trajectory privacy with Euclidean distance.
- Under the trajectory privacy and energy consumption constraints, we formulate the game reward maximization problem, which is a mixed integer linear programming (MILP) problem. By relaxing binary variables, we derive the upper bound. We also propose the heuristic algorithm for feasible solutions, and conduct computation analysis.
- Through evaluation, we show that our proposed scheme is effective in maximizing the virtual reward of LBGs while keeping the LBG player's trajectory at least k-anonymous against tensor voting based inference attacks.

The rest of paper is organized as follows. We review the related work on location privacy and trajectory privacy in Section II. In Section III, we present the overview of our system. In Section IV, we propose a novel attack model based on tensor voting theory. In Section V, we formulate the virtual reward optimization problem under k-anonymous trajectory privacy preserving constraints, derive an upper bound and illustrate a heuristic algorithm to feasibly solve the problem. In Section VI, we conduct the performance evaluation, and analyze the obtained results. Finally, we draw conclusions in Section VII.

II. Related Work

In existing literature, there are many papers studying location privacy. Those location privacy preserving efforts can be generally classified into three categories. The first category is sending fake locations along with true locations of the user to the LBG provider, which is called dummy-based location privacy preservation [5]–[8]. In this case, users send dummy requests together with the true request, hence the attacker cannot distinguish the real location from the dummy locations. For example, in [5], the proposed spatiotemporal correlation-aware dummy-based location privacy protection scheme could prevent the location information disclosure from consecutive requests. The second category is sending a time or space obscure location to the LBG instead of the true location of the user, which is called obfuscation [9]–[11]. Some schemes belonging to this category often put the true location together with another $k-1$ dummy locations in an area in order to keep the probability of finding out the true location at $1/k$, which is called location spatial cloaking [7], [12]–[14]. Most designs with the location spatial cloaking approach use the syntactic privacy models, which are sensitive to inference attacks. In this case, this solution does not provide rigorous privacy under such situations as the source and destination of a user's trajectory may be acknowledged, for example, the user may post location through social media like Facebook. For instance, in [7], with the consideration of LBS users' side information, the authors proposed dummy-based location privacy preservation schemes which are able to achieve k-anonymity. The dummy locations were chosen based on the entropy privacy metric. The last kind of methods is the mix-zone model [15], [16], which is first proposed to be used in location privacy preservation in [17]. A mix-zone indicates that when users enter the mix-zone, they can change their pseudonym to prevent the adversary from tracking their locations.

Beyond location privacy, it is far more challenging and complicated to preserve trajectory privacy. In addition, if the trajectory of a user is exposed, the locations of the user may be known by the adversary. One popular way of trajectory protection is generating dummy trajectories. For example, in [18], two dummy-based schemes, random pattern scheme and rotation dummy generation, were proposed. The first generated dummy trajectory randomly from start to end locations and the second one rotated the original trajectory by a location along the trajectory. There is another technique to protect trajectory privacy, which is trajectory k-anonymity. In [13], the authors proposed a trajectory privacy protection scheme against semantic and re-identification attacks. In the meanwhile, the conditions of k-anonymity could be satisfied. Most work in trajectory privacy preservation only concentrates on proposing a new privacy protection framework, and have limited concern about how to quantify the privacy. Quantitative privacy metrics are in need, which helps users understand to what extent the trajectory privacy is preserved.

III. System Overview

Generally speaking, LBG players[1] tend to harvest as much virtual reward or currency as possible while playing LBGs.

[1]We use LBG players and LBG users interchangeably in this paper.

Although winning virtual reward entertains LBGs players, there exist potential risks that dishonest third-party LBG servers or eavesdropping attackers are able to analyze their location information and track their trajectories. Therefore, it is necessary to study the methods which attackers use to infer users' trajectories. In our paper, we assume the attacker uses tensor voting based inference attack to track the user. Therefore, we propose a trajectory privacy preserving scheme which is able to maximize the virtual game rewards of users. In this paper, we focus on the trajectory privacy preservation scheme and the balance of the game is out of the scope of our topic. Before we illustrate our system, we list assumptions as follows:

- We assume that the attacker is an active attacker that can access to the history data of a user in order to learn the user's living habits. Additionally, the attacker uses tensor voting based inference attack, which we will introduce in Section IV, to track the user's trajectory.
- We take time t as the timestamp for each location from a time set for the trajectory $\mathcal{T} = \{1, \ldots, t, \ldots, T\}$.
- In our work, we assume the users report each location along the true path together with the other $k-1$ candidate dummy locations on the map, which can be chosen as fake locations of the users. We assume the set of locations on the real path is $\mathcal{T}R = \{L_1, \ldots, L_i\}$. The set of candidate dummy locations is $\mathcal{D} = \{L_{d1}, \ldots, L_{dj}\}$, where dummy locations are chosen from a candidate set $\mathcal{C} = \{1, \ldots, c, \ldots, C\}$. The fake trajectory set is $\mathcal{T}R_d = \{TR_{d,1}, \ldots, TR_{d,n}\}$, where n means the n-th fake trajectory and is in the set $\mathcal{N} = \{1, \ldots, n, \ldots, k-1\}$.
- We assume the source and destination locations are publicly known, because these locations can easily be identified by others. For example, if the user is a student, in the morning, he/she should go to school from home. Moreover, the users may share their locations via the social networks such as Facebook, Instagram and so on. Therefore, these two locations are easily known to the attacker. Each of the rest of locations along the true trajectory is supposed be reported to the service with other $k-1$ dummy locations chosen from the candidate location set $\mathcal{D}$.

In Fig. 1, it is a map from PokémonFind website, which can display Pokémons locations so as to help PokémonGo players catch Pokémons. The yellow diamonds mean that there exists Pokémon. The pink triangles show that there is a gym of the game. The squares are candidate dummy locations. All of the locations talked above are in the set $\mathcal{D}$. We set different Pokémon locations as different reward values to simulate the virtual reality of the game. The Pokémons and gyms locations are included in the candidate location set. As we treat the PokémonGo player as LBG users, we would like to generate another $k-1$ trajectories to keep his trajectory k-anonymous. Besides, we would like the LBG users to have as much virtual reward as possible along the other $k-1$ trajectories.

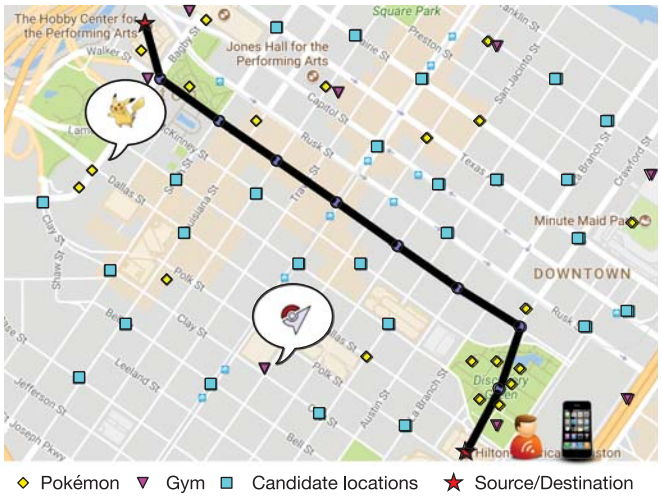


Fig. 1. A schematic of system model inspired by Pokémon Go game.

IV. Tensor Voting Based Inference Attacks

A. Outlines of Trajectory Inference Attacks via Tensor Voting

Tensor voting is an unsupervised data-driven methodology to automatically infer and group geometric objects [3], which systematically explains how to infer hidden structures like gaps and broken parts in the trace trajectory [19], [20]. As for trajectory inference attacks, the dishonest LBG servers or eavesdropping attackers may exploit tensor voting theory to infer a user's trajectory, because tensor voting has desired geometric properties such as smoothing continuous trajectories and bounding boxes with minimum registration errors.

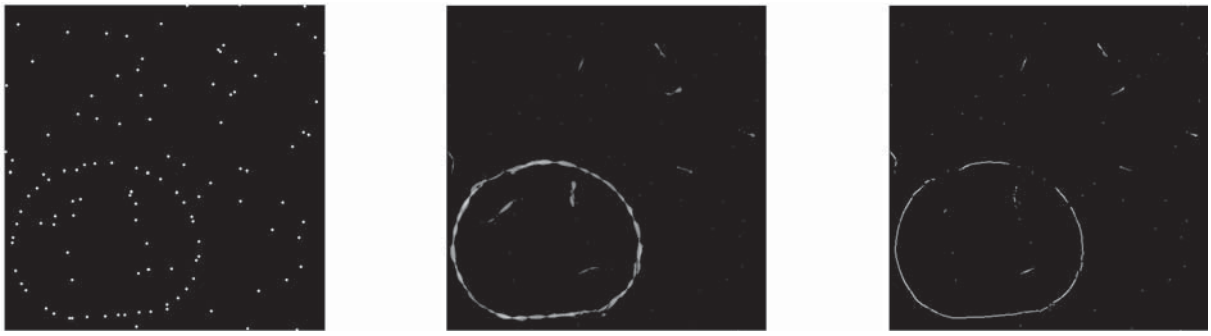
Those salient properties make tensor voting based inference attacks superior other inference attacks [4] because the adversary only needs partial/limited information to launch inference attacks via tensor voting. For example, as shown in Fig. 2, even without any timestamps, the adversary can still leverage the historical/known locations to infer the user's trajectory using tensor voting. In general, given the collected location data of the LBG user, the adversary can encode the normal space with tensor representation and mathematically infer the trajectory of the LBG user according to tensor voting theory.

In the rest of this section, we introduce the tensor voting framework in 2-D. As shown in Fig. 2(a), attackers are able to collect history locations of a user. With the tensor voting process, the outlier locations are filtered out shown in Fig. 2(b). After feature extraction, attackers can mathematically track the user's trajectory. Next, we will illustrate the approach to representing a token, which is encoded with normal space. Then, we introduce the tensor voting based inference attack procedure.

B. Second Order Representation

The structure information of an input location site can be encoded as a tensor. According to Gestalt principles [21], the exist of objects or shapes which are close enough indicates that these objects probably appear as a group. The strength of each type of visual structure, or saliency, and the preferred normal directions can be encoded within a second order symmetric non-negative definite tensor.

To begin with, we need to mathematically model the structures. In a $N-d$ space, there is a set of N orthonormal basis vectors $\hat{\mathbf{e}}_1, \ldots, \hat{\mathbf{e}}_N$, where d basis vectors from the beginning of this set span the normal space and the rest $N-d$ vectors span



(a) History locations of a LBG user. (b) After tensor voting processing. (c) After feature extraction.

Fig. 2. An illustrative example of tensor voting based inference attacks.

the tangent space. The representation of the normal space in d dimensions is

$$\mathbf{N}_d = \sum_{k=1}^{d} \hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^{\mathrm{T}}. \tag{1}$$

Therefore, the projection of a vector $\mathbf{v}$ into the 2-D normal space should be

$$\mathbf{v}_n = \sum_{k=1}^{d} \hat{\mathbf{e}}_k \left(\hat{\mathbf{e}}_k^{\mathrm{T}} \mathbf{v}\right) = \left(\sum_{k=1}^{d} \hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^{\mathrm{T}}\right) \mathbf{v} = \mathbf{N}_d \mathbf{v}. \tag{2}$$

In our work, we only consider 2-D with d equals to 2.

Normal space represents the structure types well, but it is required to know how salient the structures are in order to adequately model the structure. We encode saliency and normal spaces into a second order, symmetric, non-negative definite tensor, because the parameters are associated with the structure type. Furthermore, the second order tensor is equivalent to a 2×2 matrix, or an ellipse. The directions of two eigenvectors are the axes directions of the tensor. The major axis of the ellipse is the preferred normal orientation of a potential curve going through the location. The size of the ellipse indicates the certainty of the preferred orientation. An arbitrary second order, symmetric, non-negative definite tensor can be decomposed as:

$$\begin{aligned} \mathbf{T} &= \sum_{i=1}^{d} \lambda_i \hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^{\mathrm{T}}, \quad (d = 2) \\ &= \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^{\mathrm{T}} + \lambda_2 \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^{\mathrm{T}} \\ &= (\lambda_1 - \lambda_2)\, \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^{\mathrm{T}} + \lambda_2 (\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^{\mathrm{T}} + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^{\mathrm{T}}), \end{aligned} \tag{3}$$

where λ_i are the eigenvalues and $\hat{e}_i$ are the corresponding eigenvectors. We further define

$$s = \lambda_1 - \lambda_2, \tag{4}$$

as the saliency of the tensor. In (3), the first term refers to the stick tensor, which shows the elementary curve token with the eigenvector $\hat{\mathbf{e}}_1$ as the curve normal direction. The second term corresponds to the ball tensor that indicates a structure which has no preference of normal orientation or an intersection where two or more paths cross with each other. Therefore, if $\lambda_1 - \lambda_2$ is much larger than λ_2, it means the stick tensor is dominant

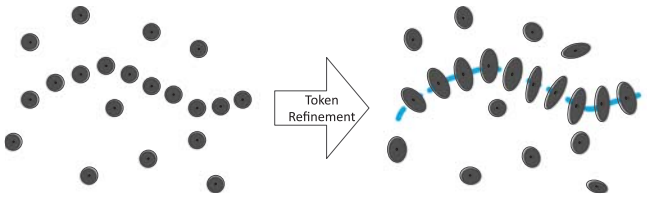


Fig. 3. Token refinement.

and infers that the curve goes through this token has a normal direction parallel to the orthonormal basis vector $\hat{\mathbf{e}}_1$. When λ_1 is approximately equal to λ_2, the tensor will become a ball tensor which shows the token is a junction or out of the structure.

C. Tensor Voting in 2-D

After the input sites have been encoded with tensors, the voting procedure is used to communicate information from each input site, or voter, to any output site, or receiver.

Analysis begins with no information at the input sites other than their locations. We create a token at each input site, according to the second order representation, initialized with a unit ball tensor indicating that no separation of the normal space from the tangent space is yet known. The first step of tensor voting, named as sparse voting, which is used to communicate information among token locations, refined tokens have encoded saliency and preferred directions of normal space at the input sites. Major and minor axes of the ellipse in Fig. 3 align with the preferred normal and tangent directions, respectively. The difference between the major and minor axis lengths represents the degree to which structure at the token is curve-like. In addition, the outliers tend to have lower saliency and are less curve-like because they are unorganized and unlikely to conspire to form a false structure. The second step of tensor voting is dense voting, which means the tokens cast vote to every neighbor location regardless of the presence of tokens. After these two steps, we can get a dense saliency figure which shows the map of saliency.

In this subsection, we use Fig. 4 as an example to illustrate the tensor voting procedure in 2-D. Stick vote is used in tensor voting to transmit information about the normal direction from a voter point $\mathcal{O}(x_1, y_1)$ to a votee point $\mathcal{P}(x_2, y_2)$. The tensors

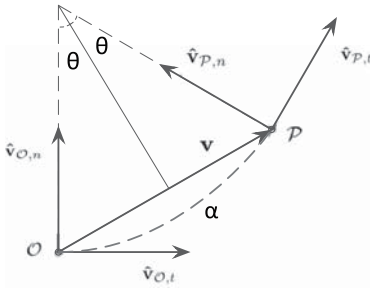


Fig. 4. Illustration of the stick vote.

of them after encoding can be represented by

$$\mathbf{T}_{\mathcal{O}} = \lambda_{\mathcal{O},1}\hat{\mathbf{v}}_{\mathcal{O},n}\hat{\mathbf{v}}_{\mathcal{O},n}^{\mathrm{T}}, \tag{5}$$

where the unit normal vector of point $\mathcal{O}$ is $\hat{\mathbf{v}}_{\mathcal{O},n}^{\mathrm{T}} = \begin{bmatrix} 0 & 1 \end{bmatrix}$, and the unit tangent vector is $\hat{\mathbf{v}}_{\mathcal{O},t}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 \end{bmatrix}$. We assume the voter and votee are connected by an arc of the osculating circle passing through them, so the normal of the votee $\mathcal{P}$ is $\hat{\mathbf{v}}_{\mathcal{P},n}$. In Fig. 4, $\mathbf{v}^{\mathrm{T}} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \end{bmatrix}$ is the vector from voter $\mathcal{O}$ to votee $\mathcal{P}$, θ is half of the central angle between $\mathcal{P}$ and $\mathcal{O}$ which is also the angle between vector $\mathbf{v}$ and vector $\hat{\mathbf{v}}_{\mathcal{O},t}$ and α is the arc length from point $\mathcal{O}$ to $\mathcal{P}$. Geometrically, we can obtain normal vector $\hat{\mathbf{v}}_{\mathcal{P},n}$ of votee $\mathcal{P}$ is

$$\hat{\mathbf{v}}_{\mathcal{P},n} = \hat{\mathbf{v}}_{\mathcal{O},n}\cos 2\theta - \hat{\mathbf{v}}_{\mathcal{O},t}\sin 2\theta = \begin{bmatrix} -\sin 2\theta \\ \cos 2\theta \end{bmatrix}, \tag{6}$$

where half of the central angle θ is

$$\theta = \arcsin \hat{\mathbf{v}}^{\mathrm{T}}\hat{\mathbf{v}}_n = \arcsin \frac{(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}, \tag{7}$$

and arc length α is

$$\begin{aligned} \alpha &= \frac{\|\mathbf{v}\|\theta}{\sin\theta} \\ &= \frac{[(x_2 - x_1)^2 + (y_2 - y_1)^2]\arcsin\frac{(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}}{y_2 - y_1}. \end{aligned} \tag{8}$$

During the voting procedure, votes are not cast equally from a token to another. The vote will attenuate with distance, for the sake of reducing the influence between unrelated tokens. Additionally, the voter will not cast any vote to a receiver which is at an angle larger than $\pi/4$ with respect to the tangent of the osculating circle at the voter. The attenuation function can be given empirically,

$$DF(\alpha, \kappa, \sigma) = e^{-\left(\frac{\alpha^2 + c\kappa^2}{\sigma^2}\right)}, \tag{9}$$

where κ is the curvature that can be found as

$$\kappa = \frac{2\sin\theta}{\|\mathbf{v}\|} = \frac{2(y_2 - y_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \tag{10}$$

c is the penalty for curvature and the σ is the only parameter that the user can change to set the scale of voting. The parameter c is also used to control the degree of decay with curvature, which is set to: $c = \frac{-16\log(0.1)(\sigma - 1)}{\pi^2}$. We can find that the attenuation function is a normal distribution function which is corresponding to a real number. The stick vote cast from voter $\mathcal{O}$ to votee $\mathcal{P}$ is as the following,

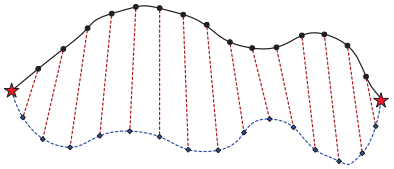


Fig. 5. Euclidean distance based trajectory privacy metrics.

$$\mathbf{V}_{\mathcal{O},\mathcal{P}} = DF(\alpha, \kappa, \sigma)\hat{\mathbf{v}}_{\mathcal{P},n}\hat{\mathbf{v}}_{\mathcal{P},n}^{\mathrm{T}}, \tag{11}$$

which is also a stick tensor. Finally, stick votes received at a votee $\mathcal{P}$ are the sum of votes cast by all the input tokens. We assume that there are k locations in a set $\mathcal{K}$ on the map. The votes received by a votee $\mathcal{P}$ can be represented as

$$\mathbf{V}_{\mathcal{P}} = \sum_{x \in \mathcal{K}} \mathbf{V}_{x,\mathcal{P}}, \tag{12}$$

where $\mathbf{V}_{x,\mathcal{P}}$ is the vote point x cast to point $\mathcal{P}$. Because the vote is also a stick tensor, equation (12) can be decomposed by (3) as following

$$\begin{aligned} \mathbf{T}_{\mathcal{P}} = {} & (\lambda_{\mathcal{P},1} - \lambda_{\mathcal{P},2})\hat{\mathbf{v}}_{\mathcal{P},n}\hat{\mathbf{v}}_{\mathcal{P},n}^{\mathrm{T}} \\ & + \lambda_{\mathcal{P},2}(\hat{\mathbf{v}}_{\mathcal{P},n}\hat{\mathbf{v}}_{\mathcal{P},n}^{\mathrm{T}} + \hat{\mathbf{v}}_{\mathcal{P},t}\hat{\mathbf{v}}_{\mathcal{P},t}^{\mathrm{T}}). \end{aligned} \tag{13}$$

V. Virtual Reward Maximization With Trajectory Privacy Preservation

In this section, we demonstrate our virtual reward maximization scheme with trajectory privacy against tensor voting based inference attacks. In Subsection V-A, we will propose our formulation for game reward maximization problem. Because the formulated problem is a mixed integer linear programming (MILP) problem, we will give the upper bound for the problem in Subsection V-B. So as to solve the formulated problem efficiently and effectively, we will manifest a heuristic algorithm for the feasible solution and analyze the complexity of the algorithm in Subsection V-C.

A. Game Reward Maximization Problem

We assume the user is a LBG player. For the purpose of defending the tensor voting based inference attack, we propose our scheme shown as Fig. 1. The user reports to the game service each true location point along the actual trajectory together with the other $k - 1$ dummy locations.[2] Based on the tensor voting analysis, we are supposed to make sure that the saliency of dummy tensors are sufficiently large to form fake trajectories. With tensor voting based inference attack, the attacker cannot distinguish fake trajectories from the actual one. In addition, the battery cost of the player's mobile device is considered as a constraint. Hence, we formulate the game reward maximization problem, while preserving the player's trajectory privacy.

[2]We assume that the LBG user will send out the true location together with $k - 1$ dummy locations simultaneously to obfuscate the adversary.

1) Method of Choosing Candidate Locations: As for a PokémonGo player, the candidate locations can be chosen from Pokémon locations, gyms of the game or other dummy locations. In Fig. 1, the squares, diamonds and triangles are regarded as the candidate locations. As we illustrate in Section III, we assume the set of candidate dummy locations is $\mathcal{D}$. The dummy locations are chosen from a candidate set, which is $\mathcal{C}$. The time set is $\mathcal{T}$. For choosing the candidate location, we denote

$$w_j^t = \begin{cases} 1, & \text{if } L_{dj} \text{ is chosen at t,} \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{for} \quad \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{C}} w_j^t \geq (k-1)\cdot(T-2)$$

$$\text{and} \quad \sum_{j\in\mathcal{C}} w_j^t \geq k-1, \tag{14}$$

where T is the total number of time slots of the whole trajectory. Like we illustrated before, we assume the source and destination locations are published, to guarantee the k-anonymity, the sum of the selected candidate location should be larger $(k-1)\cdot(T-2)$. Moreover, during one time slot, more than $k-1$ candidate locations from the set can be chosen to satisfy k-anonymity level, which is shown as (14).

2) Euclidean Distance: In order to quantity the trajectory privacy in a mathematical way, we define the location along the true trajectory at timestamp t is L_i^t, and similarly the dummy location along the n-th fake path at timestamp t is $L_{dj,n}^t$. The location L_i^t can be represented as a triple-tuple (x_i, y_i, t). Consequently, we can get the Euclidean distance between the two locations at the same timestamp as follows,

$$Eu(L_i^t, L_{dj,n}^t) = \sqrt{(x_i^t - x_{dj,n}^t)^2 + (y_i^t - y_{dj,n}^t)^2}. \tag{15}$$

After processing the locations with our scheme we illustrated in Section III, the adversary is able to get k sets of locations, which are the dummy trajectories $\mathcal{TR}_{d,n}$. In our work, the source and destination locations are overlapped by true and dummy trajectories, and hence we can define the trajectory privacy as

$$TP(\mathcal{TR}_{d,n}, \mathcal{TR}) = \sum_{t=2}^{T-1} Eu(L_i^t, L_{dj,n}^t). \tag{16}$$

So as to preserve trajectory privacy of the player, the difference between each fake trajectory and the true trajectory should be larger than a threshold value, which is shown in (17).

$$TP(\mathcal{TR}_{d,n}, \mathcal{TR}) \geq TP_{TH}. \tag{17}$$

Here, the threshold TP_{TH} is supposed to be set not that large, hence different trajectories cannot be distinguished, in the meanwhile the true trajectory will not be found apart from the other $k-1$ trajectories.

3) Energy Consumption Constraint: When we generate the plain text and transmit it through the wireless network, there is energy consumption. Because our scheme is used in mobile devices and there is a limited power usage, we need to consider the power usage when processing the our scheme. The extra energy consumption is from generation and transmission of the dummy locations. It is supposed to guarantee that after carrying out the scheme, the battery life time is still long enough for other use. Therefore, the energy consumption constraint is shown as follows:

$$Q - \sum_{j\in\mathcal{C}} w_j^t \cdot P_E > Q_{TH}, \tag{18}$$

where P_E is the energy consumption of generating and transmitting one dummy location, Q is the original battery capacity and Q_{TH} is the threshold of the rest of capacity after conducting the scheme.

4) Game Reward: In our proposed scheme, we would like to maximize the total game reward when the user if playing location-based games, such as PokémonGo. According to the real game data, we set different Pokémon locations with different reward values, which are represented as r_j with $j \in \mathcal{C}$. Accordingly, the total reward of the game after applying our scheme is shown as

$$R = \sum_{j\in\mathcal{C}} (r_j \cdot w_j^t). \tag{19}$$

5) Tensor Voting Constraint: As illustrated in Section IV, we take tensor voting analysis to launch trajectory inference attacks. So as to hide the true path, after processing with tensor voting, the saliency of dummy locations should be larger than an upper bound threshold value s_{TH_H}. From Subsection IV-C, we can obtain the tensor of location L after voting procedure can be represented as following

$$\begin{aligned} \mathbf{T}_L = {} & (\lambda_{L,1} - \lambda_{L,2})\hat{\mathbf{v}}_{L,n}\hat{\mathbf{v}}_{L,n}^{\mathrm{T}} \\ & + \lambda_{L,2}(\hat{\mathbf{v}}_{L,n}\hat{\mathbf{v}}_{L,n}^{\mathrm{T}} + \hat{\mathbf{v}}_{L,t}\hat{\mathbf{v}}_{L,t}^{\mathrm{T}}). \end{aligned} \tag{20}$$

As defined in Subsection IV-B, the saliency of the tensor of location L should be

$$s_L = \lambda_{L,1} - \lambda_{L,2}. \tag{21}$$

Accordingly, the saliency of the locations along the dummy trajectory can be represented as $s_{L_{dj}}$. Hence, with the aim of mixing the true and dummy trajectories, the saliency of each dummy location needs to satisfy

$$w_j^t \cdot s_{L_{dj}} \geq w_j^t \cdot s_{TH_H} \quad (j \in \mathcal{C}). \tag{22}$$

In addition, based on tensor voting theory, we can also get the maximum distance d_{max} from the voter to the votee. Normally, if the vote cast from the voter only has 1% of the voter's saliency, as $e^{-(d_{max}{}^2/\sigma^2)} = 0.01$, the votee is considered far from the voter.

6) Problem Formulation: Given the proposed trajectory privacy preservation approach above, the formulation for the LBG user's reward maximization problem can be described as

follows,

$$\begin{aligned} \text{Maximize}: \quad & R \\ \text{s.t.}: \quad & (17), (18), (19), (22) \\ & w_j^t = \{1, 0\} \quad (j \in \mathcal{C}), \\ & \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{C}} w_j^t \geq (k-1) \cdot (T-2), \\ & \sum_{j \in \mathcal{C}} w_j^t \geq k - 1, \end{aligned} \tag{23}$$

where w_j^t is the optimization variable and all of other parameters are constants illustrated before. Trajectory privacy is guaranteed by the constraint (17), which is obtained by Euclidean distance between locations on true and fake trajectories.

B. The Upper Bound for Trajectory Privacy Optimization

The formulated trajectory privacy maximization problem is an MILP problem, which is NP-hard to solve in general [22], [23]. The complexity of the optimization results from the integer parameter w_j^t. We can relax the binary variable w_j^t from $\{0, 1\}$ to real numbers in $[0, 1]$, according to the methodologies in [23]. In this case, the complexity of this optimization problem will be reduced obviously. After relaxing the integer variables, we can explore an upper bound for the formulated problem. As a result, the MILP problem is converted into a linear programming (LP) problem, which can be obtained in polynomial time and solved using CPLEX [24].

C. The Heuristic Algorithm for Feasible Solutions

As illustrated in Subsection V-B, we are able to get the upper bound for the proposed problem as the benchmark, nevertheless we still explore for an effective and feasible solution. In this subsection, we will describe our heuristic algorithm to solve this optimization problem.

It is obvious that if all the dummy locations in the set $\mathcal{D}$ are determined to be chosen or not, which means all of the w_j^t-variables are decided, the proposed trajectory privacy maximization problem will become an LP problem. In this case, we first relax binary w_j^t-variables to $0 \leq w_j^t \leq 1$, and hence the problem is converted to an LP problem. This LP problem can be solved by several mathematical tools, so we are able to achieve the feasible solution that every w_j^t-variable should be a decimal value between 0 and 1. All w_j^t with decimal values are put into a set W_j^t. If all of the fractional values are smaller than 0.5, we fix the minimal value of w_j^t, represented as w_n^t, to 0. Otherwise, there should be a maximal value of w_j^t values, which is assumed to be represented as w_m^t, and then we set w_n^t to 1. Subsequently, we can relax the rest of w_j^t-variables and perform an updated LP problem as above. The procedure of the heuristic algorithm is shown in Algorithm 1. Upon iterations of solving the updated LP problem, we can fix all the w_j^t-variables. After fixing w_j^t-variables, the original MILP is converted into an LP and can be feasibly solved.

Algorithm 1: Relex-and-Fix Heuristic Algorithm.

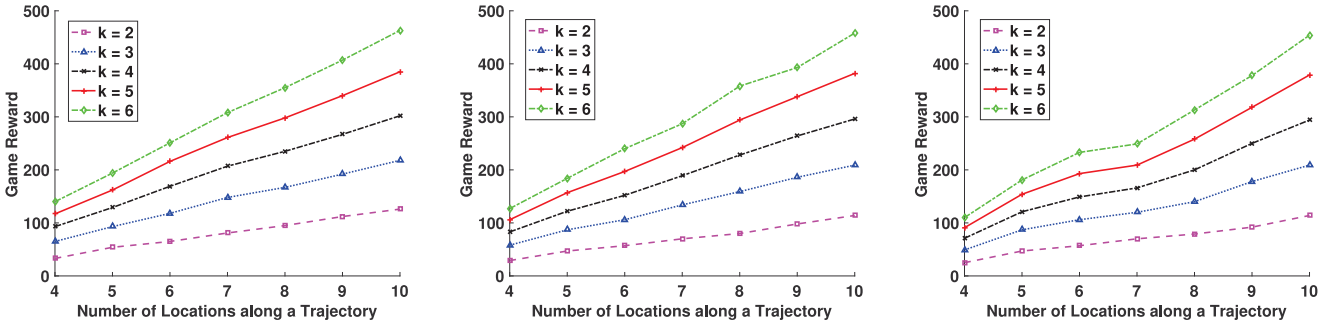
```
Data: w_j^t LP feasible values
W_j^t ← set of all w_j^t with fractional values;
while W_j^t ≠ ∅ do
    if all fractional values in W_j^t < 0.5 then
        fix the minimum w_n^t to 0;
        W_j^t \ w_n^t;
        reformulate and solve the new relaxed LP
          problem with fixed w-variables;
    else
        fix the maximum w_m^t to 1;
        W_j^t \ w_m^t;
        reformulate and solve the new relaxed LP
          problem with fixed w-variables;
    end
end
return all fixed w_j^t-values;
```

After the description of the heuristic algorithm for the proposed problem, we analyze and compare the complexity of the optimization solution of the MILP problem formulation and the feasible solution with the heuristic algorithm. In this MILP problem, there is one binary variable w_j^t for $j \in \mathcal{C}$. Therefore, the possible combinations of w_j^t is 2^C. As said, considering all of w_j^t are fixed, the MILP problem will become an LP problem. According to [25], we can find that the intrinsic computational complexity of an LP problem is $O(A^3 \cdot L)$, where A is whether the number of constraints or variables in the problem depending on which one is larger, and L is the number of binary bits required to store the data, which is the input length of a situation of the proposed problem. The number of variables is C^2, which is larger than the number of constraints C, the complexity of solving this LP problem is $O(C^6 \cdot L)$. Consequently, the computational complexity for the optimal solution of the proposed MILP formulation is $O(2^C \cdot C^6 \cdot L)$. Now, we continue to analyze the computational complexity of our heuristic algorithm. As illustrated before, we relax and fix the w_j^t-variable by iterations. In order to determine all the w_j^t-variables, we repeat doing iteration. The complexity for the iteration procedure is $O(C)$ and the complexity for the LP problem is $O(C^6 \cdot L)$, which results in the overall complexity is $O(C \cdot C^6 \cdot L)$. Obviously, the computational complexity is significantly reduced compared with the optimal solution with complexity $O(2^C \cdot C^6 \cdot L)$.

VI. Performance Evaluation

A. Simulation Setup

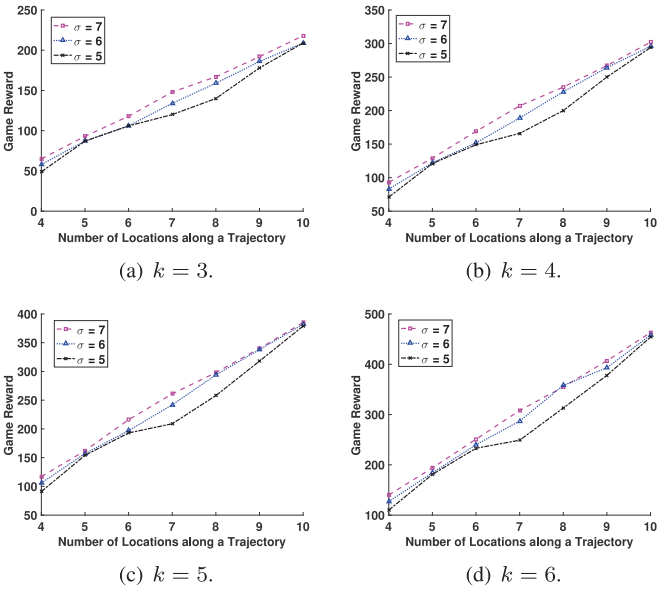
In the simulation, we used one user's GPS trajectory data from GeoLife dataset [26]. GeoLife dataset was collected by a project of Microsoft Research Asia from 182 users over three years. In the project, users reported their location to the service provider very frequently by every three or five seconds. In addition, the users' location information was recorded in the dataset by tuples of latitude, longitude and timestamp. We randomly choose one user's partial trajectory from the dataset and



(a) $\sigma = 7$ (b) $\sigma = 6$ (c) $\sigma = 5$

Fig. 6. Game reward with constant σ value and different k values.

evaluate the performance of our proposed scheme. We extract 10 locations of the selected user's trajectory every ten seconds and grant 100 dummy locations which are distributed around this trajectory area. By analyzing the extracted data, the user was walking from the source location to destination location. The speed that people tend to choose to walk is the preferred walking speed of human. Most people's preferred walking speed is around 1.4 m/s. However, some people's walking speed can also achieve to 2.5 m/s for a short distance [27]. As illustrated in Subsection IV-C, the decay function is a normal distributed function. To make the vote cast from other location higher than 1% of the voter's saliency, the maximum distance, say d_{max}, between two locations should satisfy $e^{-(\frac{d_{max}^2}{\sigma^2})} = 0.01$, which can be simplified as $d_{max} \approx 3\sigma$. Compared with the walking speed of human, we are able to set the tensor voting parameters σ as 5, 6 or 7. Given σ values, the maximum distance between two selected candidate locations are feasible to calculate when the user is walking. In addition, we generate different reward values for different locations on a designated map from GeoLife dataset.



(a) $k = 3$. (b) $k = 4$.

(c) $k = 5$. (d) $k = 6$.

Fig. 7. Game reward with different σ values and constant k value.

B. Privacy and Performance Analysis

In our work, we generate $k - 1$ trajectories to obfuscate the attackers. With the higher k value, the higher trajectory privacy can be achieved. Since the proposed scheme is used to preserve trajectory privacy against tensor voting based attack, we need to analyze LBG users' privacy preservation by launching this kind of attack. As there are two parameters σ and k in our scheme when optimizing the problem, we first evaluate the relation between both parameters. In the tensor voting procedure, σ is a parameter to control the scale of voting, which affects the saliency value of locations. The parameter k indicates that there are k trajectories in the region, among which $k - 1$ trajectories are generated by candidate dummy locations. As shown in Fig. 6, we set σ as fixed values and compare the game reward with different k values along trajectories with different lengths. We find that game reward is higher with longer trajectories and larger k values, because of the increased number of locations. Additionally, with smaller σ value, the optimized game reward decreases as well, since it has influences on tensor voting constraint of the formulation, when the voting scale σ is small. In Fig. 7, the k value is fixed and we evaluate the relation between σ and game reward. The results show that the differences of game reward between $\sigma = 7$ and $\sigma = 6$ are negligible. From the results shown in Fig. 6 and Fig. 7, we observe that game reward grows with higher k value and when trajectory gets longer, the impact of parameter σ can be ignored. The simulation results show that our scheme can preserve k-anonymous trajectory privacy against tensor voting based inference attack, while the LBG player is able to achieve more virtual game reward.

Moreover, we compare the performance of the proposed scheme with random and rotation [18] schemes. In the random scheme, the source and destination locations are the same along true and dummy trajectories. We randomly choose dummy locations from the same candidate dummy location dataset as used in our scheme to generate fake trajectories which is able to satisfy the requirement of human walking speed constraints as described before. Since the source and destination locations are public, we abuse the rotation scheme a little bit to fit this scenario. Here, only the locations except source and destination locations along true trajectory are rotated and the rotation point is the center of the trajectory. Furthermore, we limit the rotation angle in order to keep the distance between locations along a trajectory to satisfy the walking speed of human as well.

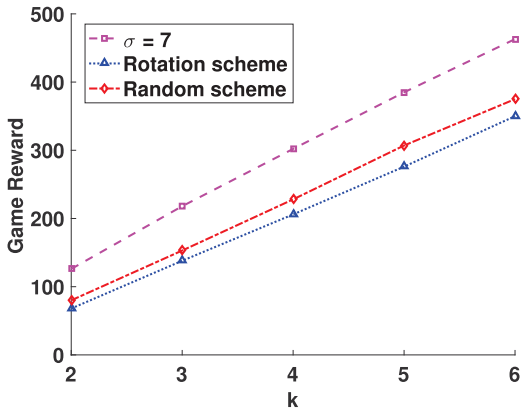


Fig. 8. Impact of k values on game reward among rotation, random, and our scheme with $\sigma = 7$.

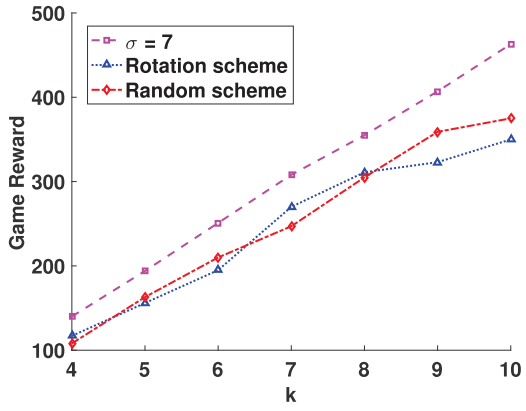


Fig. 9. Impact of trajectory length on game reward among rotation, random, and our scheme with $\sigma = 7$.

Through simulations, we compare the performances of random, rotation and our scheme with $\sigma = 7$. In Fig. 8, we evaluate the impact of k values when there are 10 locations along the true trajectory. The result shows that our proposed scheme is superior to either random or rotation schemes in terms of virtual reward harvesting. In Fig. 9, we evaluate the impact of trajectory length with $k = 6$. With longer trajectory, the virtual reward achieved by our scheme is also much higher than the other two schemes.

VII. Conclusion

In this paper, our motivation is to address the LBG players' trajectory privacy problem. We first introduced a novel tensor voting based inference attack. To thwart this inference attack, we proposed a dummy-based k-anonymous trajectory privacy preserving approach, which is able to quantify trajectory privacy by Euclidean distance and meet k-anonymity requirements. Since our target is on LBG players, the virtual game reward is considered in the problem. Specifically, we have mathematically formulated the LBG users' virtual game reward maximization problem under privacy, tensor voting and hand-held devices' energy consumption constraints. Due to the NP-hardness of the formulated problem, we have developed heuristic algorithms for feasible solutions. Through extensive simulations, we have shown that the proposed scheme can effectively maximize the game reward of LBG players while keeping LBG user's trajectory k-anonymous against the tensor voting based inference attack.

References

[1] M. Serino, K. Cordrey, L. McLaughlin, and R. L. Milanaik, "Pokémon go and augmented virtual reality games: A cautionary commentary for parents and pediatricians," *Current Opinion Pediatrics*, vol. 28, no. 5, pp. 673–677, Oct. 2016.

[2] S. Benford *et al.*, "The error of our ways: The experience of self-reported position in a location-based game," in *Proc. Int. Conf. Ubiquitous Comput.*, Nottingham, U.K., Sep. 2004, pp. 70–87.

[3] P. Mordohai and G. Medioni, *Tensor Voting: A Perceptual Organization Approach to Computer Vision and Machine Learning*. San Rafael, CA, USA: Morgan & Claypool, 2006.

[4] X. Pan, J. Xu, and X. Meng, "Protecting location privacy against location-dependent attacks in mobile services," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 8, pp. 1506–1519, Aug. 2012.

[5] H. Liu, X. Li, H. Li, J. Ma, and X. Ma, "Spatiotemporal correlation-aware dummy-based privacy protection scheme for location-based services," in *Proc. IEEE Int. Conf. Comput. Commun.*, Atlanta, GA, May 2017.

[6] L. Hua, J. C. S, and M. L. Yiu, "Pad: Privacy-area aware, dummy-based location privacy in mobile services," in *Proc. 7th ACM Int. Workshop Data Eng. Wireless Mobile Access*, Vancouver, Canada, Jun. 2008, pp. 16–23.

[7] B. Niu, Z. Zhang, X. Li, and H. Li, "Privacy-area aware dummy generation algorithms for location-based services," in *Proc. IEEE Int. Conf. Commun.*, Sydney, Australia, Jun. 2014, pp. 957–962.

[8] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing privacy through caching in location-based services," in *Proc. IEEE Int. Conf. Comput. Commun.*, Hong Kong, China, Apr. 2015, pp. 1017–1025.

[9] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, and X. Ma, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation," in *Proc. Int. Conf. World Wide Web*, Perth, Australia, Apr. 2017, pp. 627–636.

[10] T. Jiang, H. J. Wang, and Y.-C. Hu, "Preserving location privacy in wireless lans," in *Proc. 5th Int. Conf. Mobile Syst., Appl. Services*, San Juan, Puerto Rico, Jun. 2007, pp. 246–257.

[11] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Scottsdale, AZ, Nov. 2014, pp. 251–262.

[12] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Trans. Mobile Comput.*, vol. 7, no. 1, pp. 1–18, Nov. 2008.

[13] Z. Tu, K. Zhao, F. Xu, Y. Li, L. Su, and D. Jin, "Beyond k-anonymity: protect your trajectory from semantic attack," in *Proc. 14th Annu. IEEE Int. Conf. Sens., Commun., Netw.*, San Diego, CA, Jun. 2017.

[14] K. Vu, R. Zheng, and J. Gao, "Efficient algorithms for k-anonymous location privacy in participatory sensing," in *Proc. IEEE Int. Conf. Comput. Commun.*, Orlando, FL, Mar. 2012, pp. 2399–2407.

[15] X. Liu, H. Zhao, M. Pan, H. Yue, X. Li, and Y. Fang, "Traffic-aware multiple mix zone placement for protecting location privacy," in *Proc. IEEE Int. Conf. Comput. Commun.*, Orlando, FL, Mar. 2012, pp. 972–980.

[16] B. Palanisamy and L. Liu, "Mobimix: Protecting location privacy with mix-zones over road networks," in *Proc. IEEE 27th Int. Conf. Data Eng.*, Hannover, Germany, Apr. 2011, pp. 494–505.

[17] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Comput.*, vol. 2, no. 1, pp. 46–55, Jan.–Mar. 2003.

[18] T.-H. You, W.-C. Peng, and W.-C. Lee, "Protecting moving trajectories with dummies," in *Proc. IEEE Int. Conf. Mobile Data Manage.*, Mannheim, Germany, May 2007, pp. 278–282.

[19] E. Pan, M. Pan, and Z. Han, "Tensor voting techniques and applications in mobile trace inference," *IEEE Access*, vol. 3, pp. 3000–3009, Dec. 2015.

[20] Z. Han, M. Hong, and D. Wang, *Signal Processing and Networking for Big Data Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[21] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. Comput.*, vol. 100, no. 1, pp. 68–86, Jan. 1971.

[22] M. R. Gary and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: WH Freeman and Company, 1979.
[23] M. Pan, P. Li, Y. Song, Y. Fang, and P. Lin, "Spectrum clouds: A session based spectrum trading system for multi-hop cognitive radio networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, Orlando, FL, USA, Mar. 2012, pp. 1557–1565.
[24] ILOG IBM, "Cplex optimization studio," 2014. [Online]. Available: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer
[25] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*. Hoboken, NJ, USA: Wiley, 2011.
[26] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proc. 18th Int. Conf. World Wide Web*, Madrid, Spain, Apr. 2009, pp. 791–800.
[27] R. C. Browning, E. A. Baker, J. A. Herron, and R. Kram, "Effects of obesity and sex on the energetic cost and preferred speed of walking," *J. Appl. Physiol.*, vol. 100, no. 2, pp. 390–398, Feb. 2006.

Xinyue Zhang (S'17) received the B.E. degree in communication engineering from Beijing Jiaotong University, China, in 2016 and the B.Sc. degree in electronic engineering from KU Leuven, Leuven, Belgium, in 2016. She is currently working toward the Ph.D. degree at the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA. She has been a Research Assistant with the Cognitive Radio Networking, Cybersecurity, and Cyber-Physical System Laboratory since 2017. Her research interests include cognitive radio networks and wireless security.

Jingyi Wang (S'16) received the B.S. degree in physics from Nankai University, Tianjin, China, in 2012 and the M.S. degree in electrical and computer engineering from Auburn University, Auburn, AL, USA, in 2015. Since August 2015, she has been working toward the Ph.D. degree at the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA. Her research interests include privacy preservation of cognitive radio networks, distributed spectrum trading, and wireless big data privacy. Her work on cognitive radio network won the Best Paper Award in Globecom 2017.

Yong Li (M'09–SM'16) received the B.S. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2007 and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2012. He is currently a Faculty Member with the Department of Electronic Engineering, Tsinghua University.

Dr. Li was the General Chair, TPC Chair, TPC Member for several international workshops and conferences, and he is on the editorial board of two IEEE journals. His papers have total citations, more than 4600. Among them, ten are ESI highly cited papers in computer science, and four received conference Best Paper (run-up) awards. He received IEEE 2016 ComSoc Asia-Pacific Outstanding Young Researchers and Young Talent Program of China Association for Science and Technology.

Riku Jäntti (M'02–SM'07) received the M.Sc. degree (with distinction) in electrical engineering in 1997 and the D.Sc. degree (with distinction) in automation and systems technology in 2001, both from the Helsinki University of Technology (TKK), Espoo, Finland. He is an Associate Professor (tenured) in Communications Engineering and the Head of the Department of Communications and Networking, Aalto University School of Electrical Engineering, Finland. Prior to joining Aalto (formerly known as TKK) in August 2006, he was a Professor pro tem with the Department of Computer Science, University of Vaasa. He is an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is also IEEE VTS Distinguished Lecturer (Class 2016). His research interests include radio resource control and optimization for machine type communications, cloud based radio access networks, spectrum and co-existence management and RF inference.

Miao Pan (S'07–M'12–SM'18) received the B.Sc. degree in electrical engineering from the Dalian University of Technology, Dalian, China, in 2004, the M.A.Sc. degree in electrical and computer engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2007, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2012, respectively. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA. He was a recipient of NSF CAREER Award in 2014. His research interests include cognitive radio networks, cybersecurity, and cyber-physical systems. His received Best Paper awards in VTC 2018, Globecom 2017, and Globecom 2015, respectively. He is an Associate Editor for IEEE Internet of Things Journal from 2015 to 2018. He is a member of ACM.

Zhu Han (S'01–M'04–SM'09–F'14) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 1997 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, Boise, ID, USA. He is currently a Professor with the Department of Electrical and Computer Engineering as well as with the Department of Computer Science, the University of Houston, Texas, USA. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He received the NSF CAREER Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of communications systems (Best Paper Award in IEEE JSAC) in 2016, and several Best Paper awards in IEEE conferences. He is currently an IEEE Communications Society Distinguished Lecturer.