IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 1, 2018, accepted November 27, 2018, date of publication December 3, 2018,
date of current version December 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2884516

TPP: Trajectory Privacy Preservation Against Tensor Voting Based Inference Attacks

XINYUE ZHANG[1], (Student Member, IEEE), JINGYI WANG[1], (Student Member, IEEE), MINGLEI SHU[2,3], (Member, IEEE), YINGLONG WANG[2,3], (Member, IEEE), MIAO PAN[1], (Senior Member, IEEE), AND ZHU HAN[4,5], (Fellow, IEEE)

[1]Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204, USA
[2]Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China
[3]Shandong Provincial Key Laboratory of Computer Networks, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China
[4]University of Houston, Houston, TX 77004, USA
[5]Department of Computer Science and Engineering, Kyung Hee University, Seoul 02447, South Korea

Corresponding author: Minglei Shu (Smlsmile1624@163.Com)

This work was supported by the U.S. National Science Foundation under Grants CNS-1801925, CNS-1343361, CNS-1350230 (CAREER), CNS-1717454, CNS-1646607, CNS-1702850, CNS-1731424, and ECCS-1547201.

ABSTRACT The popularity of mobile devices with global positioning system (GPS) has boosted various wireless location-based services (LBSs). Certain honest-but-curious or even dishonest LBS servers may learn the users' trajectories from location trace files, and the users' privacy can be compromised. In this paper, we propose a quantitative approach to model trajectory inference attacks via tensor voting, which can be widely applied in computer vision and machine learning as a perceptual organization. To counter the tensor voting based attacks, we propose a novel trajectory privacy preservation TPP scheme, in which LBS users will intentionally generate dummy trajectories to obfuscate LBS servers. Meanwhile, the LBS users have the option to disclose their trajectories to trustworthy parties (e.g., users' parents) by sending those parties a few more encrypted locations. Considering the power constraint of hand-held mobile devices, we mathematically formulate the trajectory privacy preservation problem into a mixed integer linear programming optimization problem and propose the algorithms for optimizing solutions. Through simulations and analysis, we show that the proposed scheme can effectively preserve LBS users' trajectory privacy against tensor voting-based inference attacks with limited power consumption.

INDEX TERMS Tensor voting, trajectory privacy, obfuscation, inference attacks.

I. INTRODUCTION

The last decade has witnessed the exploding growth in the quantity and capability of consumer mobile devices such as smartphones, tablets, etc., and the proliferation of wireless services. With the advance and commercial use of global positioning system (GPS) technology, smartphones and tablets feature sensors that can pinpoint users' locations, which can allow the location-based services (LBSs) to use users' whereabouts in a variety of ways. Actually, the LBSs do more than just tell us about exactly where we are. They offer useful features based on our location from location-based discovery tools and smart search (e.g., Foursquare, Yelp, Glympse, Detour, Gowalla, Shopkick, etc.) to games and exercise tracking (e.g., Pokémon Go, Ingress, SCVNGR, etc.). For example, Foursquare encourages users to check-in at locations in return for virtual badges and points. It also helps users keep up with friends, discover what is nearby, save money and unlock deals. According to the study by Kantar TNS, LBS users are increasingly using services to enrich their daily lives, with 22% using LBSs to find their friends nearby, 26% to find restaurants and entertainment venues, 19% to check the public transport schedules, 8% to book a taxi, and 13% to find a deal or special offer.

Although LBSs bring joy and convenience to people, they also raise serious privacy concerns. Currently, most LBSs require the user's hand-held device to periodically report the location information to the service provider, and the location data will be stored in the database/servers of the LBS provider. Following this mechanism, a dishonest third-party service provider may have very chances to leverage the user's reported locations and analyze the rich trace files to infer the trajectory of the user. With the exposure of trajectory,

the users not only lose their privacy but also are vulnerable to various attacks, even some serious physical attacks. For instance, if a celebrity registers for some dishonest LBSs, he/she can be easily tracked by the paparazzi in the digital world. Correspondingly, his/her next location can be inferred, so that his/her privacy in the real world will be invaded. Another real life example happened in Missouri, July, 2016 is that 11 Pokémon Go players have been ripped off because of playing this location based game, as thieves learned their trajectory and lured those victims to remote areas outside of St. Louis. Besides robbery, it is not hard to imagine that there might be more serious crimes such as sexual assault, kidnapping, murder, assassination, etc. targeting specific victims at selected locations, due to the disclosure of users' trajectory privacy.

To avoid those issues, it is worthy to study how the dishonest service provider analyzes the location data, and infer the users' trajectories. The emerging location data has provided opportunities for trajectory prediction [1], [2]. It is necessary to innovate a scheme to preserve the trajectory privacy of LBS users. On the other hand, users sometimes would like to intentionally disclose their trajectory to trustworthy parties (e.g., their parents, family members, close friends or even some well-known LBS providers). Therefore, it is also important to satisfy those requirements of users with the proposed trajectory privacy-preserving scheme. Moreover, as the scheme is applied on a mobile device, the energy consumption should be considered as a constraint. However, most existing trajectory privacy preservation works [3]–[5] have limited concerns about this seemingly paradoxical but practical requirements of LBS users. Besides, there is a lack of quantitative approaches to analyze either inference attacks or the countering privacy preservation measures.

Aiming to address those challenges, we novelly leverage tensor voting techniques [6] to quantitatively model and analyze trajectory inference attacks. To thwart tensor voting based inference attacks, we propose a new trajectory privacy preserving TPP scheme, which satisfies the ''paradoxical'' requirements of LBS users with limited energy consumption of their smart devices. Our salient contributions are summarized as follows:

- In this paper, we consider the LBS users' trajectory privacy when enjoying the service. We propose a novel trajectory attack model based on tensor voting theory. At the meanwhile, a trajectory privacy preservation scheme TPP is offered to counter this kind of trajectory inference attack.
- In order to make progress on the privacy quantification of the performance of the proposed trajectory privacy-preserving solutions, we formulate the trajectory privacy maximization problem with Euclidean distance. Since tensor voting based trajectory inference attack is considered as the attack model, the tensor voting requirement is set as a constraint.
- In our system, the user can selectively send his/her true trajectory to the trustworthy parties. The user intuitively reports dummy locations along a fake trajectory to the service provider and some critical locations along the true trajectories will be put into a pre-processed location data set. The data set will be encrypted and transmitted to the trustworthy parties. The trustworthy parties are able to decrypt the data set, put the points back to the original trajectory and derive the actual path of the user, while the attacker cannot infer the user's true path.
- Since TPP is applied on mobile devices, we need to consider about the energy limit problem. In our work, we minimize the energy usage to prevent the mobile devices from consuming battery fast.
- Contrasted with the previous works about dummy-based trajectory privacy preservation, we evaluate and compare the performance by simulations.

The rest of paper is organized as follows. We review the related work on location 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 give the formulation of the privacy maximization problem, derive an upper bound of it and illustrate a heuristic algorithm to feasibly solve the problem. We formulate the energy cost minimization problem in Section VI. In Section VII, we analyze the performance evaluation. Finally, we draw conclusions in Section VIII.

II. RELATED WORK

In the existing literature, there are a great number of papers studying location privacy. One general location privacy preservation mechanism is obfuscation [7]–[9], which is implementing by sending a time or space obscure location to the LBS instead of the true location of a user. Some schemes of this method put the true location and 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 [10]–[12]. Most schemes of the location spatial cloaking methods use the syntactic privacy models, which are sensitive to inference attacks. Another location privacy preservation mechanism is sending fake locations along with true locations of the user to the LBS provider, which is called dummy-based [13]–[15]. In this case, users will send a dummy request together with the true request, and hence the attacker cannot distinguish the real location from the dummy location. Moreover, there are also some works based on differential privacy models to protect location privacy [16], [17]. Moreover, some location privacy preserving schemes are using the mix-zone model [18], [19], which is first proposed to be used in location privacy preservation in [20]. A mix-zone indicates that when users enter the mix-zone, they can change their pseudonym to prevent adversary from tracking their locations. The challenge of preserving location privacy is that some frameworks need a trusted third-party anonymizer to pre-process the location data, which is not always possible and can also involve privacy issues.

Beyond the location privacy, it is challenging and complicated to preserve trajectory privacy. Furthermore, if the trajectory of a user is exposed, the locations of the user will be known by the adversary. The most popular way of trajectory protection is generating dummy trajectories. For example, in [3] and [4], two dummy-based schemes, random pattern scheme and rotation dummy generation, are proposed. The first generates dummy trajectory randomly from start to end locations and the second one rotates the original trajectory by a location along the trajectory. There is another technique to protect trajectory privacy, which is trajectory k-anonymity [5]. Most work in trajectory privacy preservation only concentrates on proposing a new privacy protection framework, and have limited concern about qualifying the privacy mathematically. While in a practical way, it is essential to obtain privacy qualification with metrics. In other words, after applying the privacy-preserving framework, it should be clear to understand to what extent the preservation of the trajectory privacy is guaranteed.

III. SYSTEM DESCRIPTION OF TPP

In this section, we demonstrate the problem statements and notations. Under the non-interactive model of the privacy preservation framework, which uses the learning algorithm, we propose the trajectory privacy preservation solution TPP against tensor voting based trajectory inference attack. TPP is based on the fact that the location information is not directly sent from GPS to the third-party server, but from the user's device. Also, the user has the control location information reporting.

Before we introduce our system, without loss of generality, we list the assumptions as follows. In our work, we assume the users do not report locations along the true path but send the candidate dummy locations to the service provider, which can be chosen as fake locations of the users. In addition, we assume the source and destination locations are public 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, the two locations are easily known to the attacker as the source and destination locations of the trajectory. However, the trajectory should not disclose to others. The rest of locations along the true trajectory will be replaced with dummy locations chosen from the candidate location set $\mathcal{D}$. The set of locations on the real path is $\mathcal{TR} = \{L_1, \cdots, L_i\}$ and the set of candidate dummy locations is $\mathcal{D} = \{L_{d1}, \cdots, L_{dj}\}$. The dummy locations are chosen from a candidate set, which is $\mathcal{C} = \{1, \cdots, c, \cdots, C\}$. We take time t as the timestamp for each location from a time set for the trajectory $\mathcal{T} = \{1, \cdots, t, \cdots, T\}$. Moreover, users can potentially select to encrypt some crucial locations on the true path and send them through a secure channel to the trustworthy party who is able to decrypt the encrypted location data set. The locations are chosen from an encrypted candidate set, which is $\mathcal{E} = \{1, \cdots, e, \cdots, E\}$. We assume that the attacker is able to get access to the history data of a user in order to learn the user's living habits. Additionally, the tensor voting analysis is treated as the trajectory inference attack model, which will be introduced in Section IV in detail.

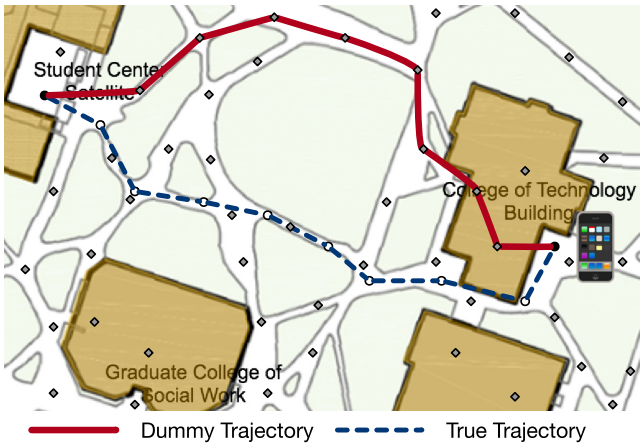


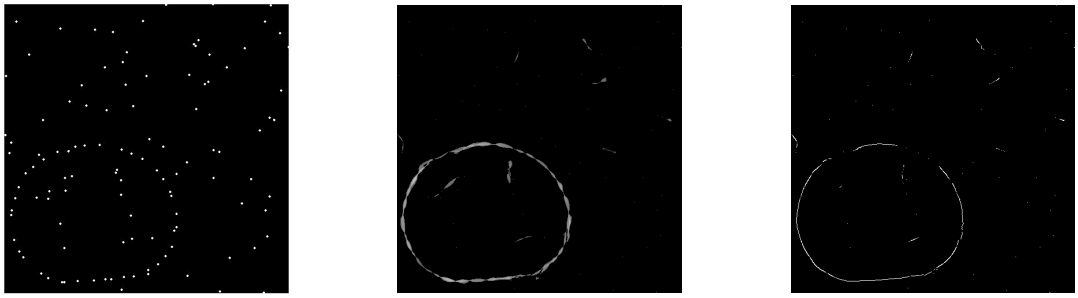
FIGURE 1. TPP network architecture.

The system overview is shown in Figure 1, which is a partial map of University of Houston from the College of Technology Building to the Student Center Satellite. The two solid circles are source and destination locations, the hollow circles are the locations along the true trajectory and the rhombi are candidate dummy locations. We assume the user is walking along the dashes line which is the true path. The user does not report the true locations but choose candidate locations, rhombi, and the fake path can be generated which is solid line on the figure. At the meanwhile, some critical location, such as turning points along the true trajectory, will be encrypted in traditional ways like Advanced Encryption Standard (AES) or Triple Data Encryption Algorithm (3DES) and send to the trustworthy party. After the trusty party decrypted the data set and put the location back to the map, the true path of the user will be shown according to the tensor voting analysis. In order to quantify and increase the trajectory privacy, we are looking forward to maximizing the differences between the true and dummy trajectories. In this case, we can protect the trajectory privacy to the greatest degree. We will further illustrate the problem formulation of maximizing trajectory privacy in Section V.

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 [6], which systematically explains how to infer hidden structures like gaps and broken parts in the trace trajectory [21]–[23]. It can be widely used in machine learning or computer vision as a perceptual organization method. As for trajectory inference



(a) (b) (c)

FIGURE 2. An illustrative example of tensor voting based inference attacks. (a) History locations of a LBS user. (b) After tensor voting processing. (c) After feature extraction.

attacks, the dishonest LBSs or eavesdropping attackers may exploit the 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 [24] because the adversary only needs partial/limited information to launch inference attacks via tensor voting. For example, as shown in Figure 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 LBS user, the adversary can encode the normal space with tensor representation and mathematically infer the trajectory of the LBS user according to the tensor voting theory.

In the rest of this section, we introduce the tensor voting framework in 2-*D*. As shown in Figure 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 Figure 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 with a tensor. According to the Gestalt principles [25], 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 $\hat{\mathbf{e}}_1, \cdots, \hat{\mathbf{e}}_N$, where d basis vectors from the beginning of this set span the normal space and the rest $N-d$ vectors span the tangent space. The representation of the normal space in d dimensions is

$$\mathbf{N}_d = \sum_{i=1}^{d} \hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T. \tag{1}$$

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

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

In our work, we only consider 2-*D*, so in the above equations, d is equal to 2.

The 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 the 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:

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

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 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.

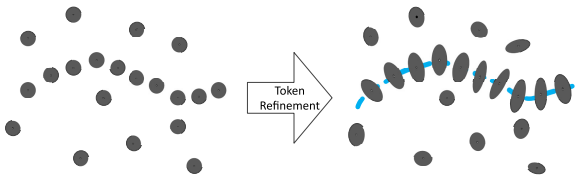


FIGURE 3. Illustration of token refinement.

Figure 3 illustrates an overview of a typical tensor voting analysis for the simple case of points in 2-*D*. 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 Figure 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 Figure 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 $O(x_1, y_1)$ to a votee point $P(x_2, y_2)$. The tensors of them after encoding can be represented by

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

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

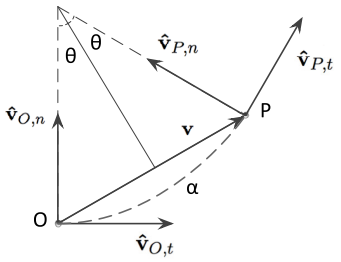


FIGURE 4. Illustration of the stick vote [21].

which is also the angle between vector $\mathbf{v}$ and vector $\hat{\mathbf{v}}_{O,t}$ and α is the arc length from point O to P. Geometrically, we can obtain normal vector $\hat{\mathbf{v}}_{P,n}$ of votee P is

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

half of the central angle θ is

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

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, in order to reduce 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 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 O to votee P is as the following,

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

which is also a stick tensor. Finally, stick votes received at a votee 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 P can be represented as

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

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

$$\mathbf{T}_P = (\lambda_{P,1} - \lambda_{P,2})\hat{\mathbf{v}}_{P,n}\hat{\mathbf{v}}_{P,n}^T + \lambda_{P,2}(\hat{\mathbf{v}}_{P,n}\hat{\mathbf{v}}_{P,n}^T + \hat{\mathbf{v}}_{P,t}\hat{\mathbf{v}}_{P,t}^T). \tag{13}$$

V. PROPOSED TRAJECTORY PRIVACY PRESERVING SCHEME AGAINST INFERENCE ATTACKS

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

A. TRAJECTORY PRIVACY MAXIMIZATION PROBLEM

In order to counter the tensor voting based inference attack, we proposed the TPP scheme shown as Figure 1. First, we let the user not report the location points along the actual trajectory, but intentionally choose candidate locations and report them which are along a dummy trajectory. Based on the tensor voting analysis, we make sure that both the saliency of dummy tensors are sufficiently large to form a fake trajectory. So with tensor voting based inference attack, the attacker believes that the user follows the dummy trajectory instead of the actual one. At the meanwhile, the user will select several specific and critical locations along the true path to be encrypted and sent to the trustworthy party. After the trustworthy party decrypts and puts these locations back to the map, the true path will appear on the map by processing with the tensor voting framework.

1) METHOD OF CHOOSING CANDIDATE LOCATIONS

As shown in Figure 1, the blue dashed line indicates the true trajectory, the red solid line indicates the fake trajectory and the two black points along the line are the source and destination locations. In order to make the attacker trust that the user follows the fake trajectory, because, sometimes, before the user is trying to fake his/her trajectories, the location of the user may be already known by the attacker and also a person cannot go too far away in a short time period, the source and destination locations are assumed to be public known locations. In Figure 1, the yellow parts are buildings and the green parts are grass or bushes. We assume that the user can go through a building and not cross through the grass or bush. We set the grey rhombi as the candidate locations, which are all along roads or close to the exit in the building. 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 $\mathcal{T}$. In order to choose the candidate location, we denote

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

$$\text{for} \quad \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{C}} w_j^t = T-2 \quad \text{and} \quad \sum_{j=1}^{C} w_j^t = 1, \tag{14}$$

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 public known, the sum of the selected candidate location should be $T-2$. Moreover, during one time slot, only one candidate location from the set can be chosen, which is shown as (11).

2) EUCLIDEAN DISTANCE

In our work, we are trying to propose a mathematical way to quantify the trajectory privacy. We define the location along the true trajectory at timestamp t is L_i^t, and similarly the dummy location along the fake path at timestamp t is L_{dj}^t. The location L_i^t can be represented as a triple-tuple (x_i, y_i, t), where x_i and y_i are the coordinates of the location. Consequently, we can get the Euclidean distance between the two locations at the same timestamp as follows,

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

After processing the locations with our scheme we illustrated in Section III, besides the public known source and destination locations, the adversary can only get one set of locations, which is the dummy trajectory $\mathcal{TR}_d = w_j^t \cdot \mathcal{D}, j \in \mathcal{C}$. In our paper, the source and destination locations are overlapped by true and dummy trajectories, and hence we can define the trajectory privacy as

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

As shown in Figure 5, we assume the black solid line and blue dashed line are two trajectories, and the red dashes lines between the two trajectories are the euclidean distances between two locations and the total length of all the red dashed line is considered as the defined trajectory privacy.

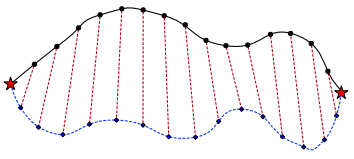


FIGURE 5. Euclidean distance based trajectory privacy metrics.

3) TENSOR VOTING CONSTRAINT

As illustrated in Section IV, we take tensor voting analysis to launch trajectory inference attacks. In order to hide the true path, after processing with tensor voting, the saliency of fake locations along the dummy trajectory should be larger

than an upper bound threshold value s_{TH_H}, and the saliency of locations except the destination and source locations along the true trajectory should be smaller than a lower bound threshold value s_{TH_L}. Likewise, after the encrypted critical points along the true trajectory has been decrypted by the trusted party and put back on the map, the saliency of locations along the true trajectory should also be larger than the 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

$$\mathbf{T}_L = (\lambda_{L,1} - \lambda_{L,2})\hat{\mathbf{v}}_{L,n}\hat{\mathbf{v}}_{L,n}^T + \lambda_{L,2}(\hat{\mathbf{v}}_{L,n}\hat{\mathbf{v}}_{L,n}^T + \hat{\mathbf{v}}_{L,t}\hat{\mathbf{v}}_{L,t}^T). \quad (17)$$

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

$$s_L = \lambda_{L,1} - \lambda_{L,2}. \quad (18)$$

Accordingly, the saliency of the locations along the dummy trajectory and the locations along the true trajectory can be represented as $s_{L_{dj}}$ and s_{L_i}. Hence, in order to hide the true trajectory and show the dummy trajectory, we need to satisfy

$$w_j^t \cdot s_{L_{dj}} \geq w_j^t \cdot s_{TH_H} \quad (j \in \mathcal{C}), \quad (19)$$

$$s_{L_i} \leq s_{TH_L} \quad (i \in \mathcal{E}). \quad (20)$$

In addition, based on tensor voting theory, we can also get the maximum distance d_{max} from the voter at which the vote cast will have 1% of the voter's saliency, as $e^{-(d_{max}{}^2/\sigma^2)} = 0.01$, since voting only takes place in a finite neighborhood.

4) PROBLEM FORMULATION

With the proposed trajectory privacy preservation system, the formulation for the trajectory privacy maximization problem can be described as follows,

$$\text{Maximize } TP(\mathcal{TR}_d, \mathcal{TR}) \quad (21)$$

$$\text{s.t.: } w_j^t = \{1, 0\} \quad (j \in \mathcal{C}), \quad (22)$$

$$\sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{C}} w_j^t = T - 2, \quad (23)$$

$$\sum_{j=1}^{C} w_j^t = 1, \quad (24)$$

$$w_j^t \cdot s_{L_{dj}} \geq w_j^t \cdot s_{TH_H} \quad (j \in \mathcal{C}), \quad (25)$$

$$s_{L_i} \leq s_{TH_L} \quad (i \in \mathcal{E}), \quad (26)$$

where w_j^t is the optimization variable, P_E, P_{TH}, s_{TH_H} and s_{TH_L} are all constant, and s_{L_i} and $s_{L_{dj}}$ is deterministic value when the only parameter σ in tensor voting procedure is given. Equations (22) (23) and (24) indicate the candidate dummy locations selection constraint. (25) and (26) specify the tensor voting constraints.

B. THE UPPER BOUND FOR TRAJECTORY PRIVACY OPTIMIZATION

The formulated trajectory privacy maximization problem is a mixed integer linear programming (MILP) problem, which is NP-hard to solve in general [26], [27]. 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 [27]. 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 [28].

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.

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 [29], 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 $O(2^C \cdot C^6 \cdot L)$.

Algorithm 1 Relex-and-Fix Heuristic Algorithm

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

VI. TRUE TRAJECTORY RECONSTRUCTION FOR TRUSTWORTHY PARTIES

Although the user prefers to fake their trajectory in order to prevent from attacking from malicious parties, the user may also desire to tell his/her true trajectory to parents or friends who are considered as trustworthy party. As we illustrated in Section III, the user can send selected locations to the trustworthy parties. However, because of sending these locations, it will cause extra communication and computation cost. In this section, we will minimize the energy cost when sending encrypted locations to trustworthy parties for true trajectory reconstruction.

A. METHOD OF CHOOSING ENCRYPTED LOCATIONS

In our work, the user can select several critical locations such as turning points to be encrypted with an encryption function $E(\cdot)$. The encrypted data set can only be decrypted by the trusted party. Similarly, the set of locations on the real path is $\mathcal{T}R$ and the locations are chosen from an encrypted candidate set $\mathcal{E}$, which are introduced in Section III. We give a function to choose the candidate location as following,

$$\delta_i^t = \begin{cases} 1, & \text{if } i \text{ is chosen to be encrypted at } t, \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

After the trustworthy party decrypts the encrypted location data, by applying the tensor voting framework, the true trajectory can be obtained by the reliable party.

B. ENERGY COST CONSTRAINT

When we encrypt 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 reduce the power usage when processing the our scheme. In our paper, the extra energy consumption is from encryption and transmission of the critical locations along the true trajectory. Therefore, we assume that the power usage need to satisfy (28), where P_E is the power cost to encrypt and transmit one location and P_{TH} is the limited power usage of the mobile device. In our paper, we use the AES scheme to encrypt the location information for example. The energy cost constraint can be represented as following

$$\sum_{i \in \mathcal{E}} \delta_i^t \cdot P_E \leq P_{TH}. \tag{28}$$

In [30], the authors designed three different experiments and gave the result data about energy consumption in a cell phone. We assume that the size a location data is 8kB and the average energy consumption of AES encryption and transmission is around 90mW according to the experiment result of [30]. Therefore, P_E in (28) is equal to 90mW. In order not to consume energy of the cell phone too fast, the energy consumption should never be higher than 1W, so P_{TH} in (28) should be 1W.

C. TENSOR VOTING CONSTRAINT

We define s'_{L_i} as the saliency of the encrypted locations along the true path after being decrypted and processed voting. The saliency should be larger than the upper bound threshold value s_{TH_H} as demonstrated in Subsubsection V-A.3, which can be represented as following

$$\delta_i^t \cdot s'_{L_i} \geq \delta_i^t \cdot s_{TH_H} \quad (i \in \mathcal{E}). \tag{29}$$

D. ENERGY COST MINIMIZATION

In order to reduce the communication and computation cost when communicating with trustworthy party, we employ the formulation as following

$$\text{Minimize} \sum_{i \in \mathcal{E}} \delta_i^t \tag{30}$$

$$\text{s.t.: } \delta_i^t = \{1, 0\} \quad (i \in \mathcal{E}), \tag{31}$$

$$\sum_{i \in \mathcal{E}} \delta_i^t \cdot P_E \leq P_{TH}, \tag{32}$$

$$\delta_i^t \cdot s'_{L_i} \geq \delta_i^t \cdot s_{TH_H} \quad (i \in \mathcal{E}). \tag{33}$$

where δ_i^t is variable, P_E, $P_T H$ and s_{TH_H} are constant, s'_{L_i} is also a constant if the only parameter σ in tensor voting is confirmed, (31) signifies the encryption of true locations constraint, (32) expresses the energy cost constraint, and (33) represents the tensor voting constraint.

VII. PERFORMANCE EVALUATION

In this section, we will analyze the security and discuss about the simulation results about our proposed trajectory privacy maximization and energy cost minimization problem.

A. SIMULATION SETUP

In the simulation, we consider there are 2 to 10 locations including the source and destination locations along the true trajectory. We grant 30 dummy locations which are distributed in a 200×50 m^2 area. We assume that users will send their locations to server every 30 seconds. 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, people's walking speed can also achieve to 2.5 m/s in a short distance [31]. As illustrated in Subsection IV-C, the decay function is a normal distributed function. In order 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 $d_{max} \approx 3\sigma$. In this case, compared with the walking speed of human, we are able to set the tensor voting parameters σ as 10, 20 and 25.

B. SECURITY ANALYSIS

The trajectory privacy is preserved by reporting dummy locations instead of true locations of the user. As illustrated in Section V, we define the trajectory privacy as the difference between true and dummy trajectory which is represented by $TP(\mathcal{TR}_d, \mathcal{TR})$. We apply the heuristic algorithm described in Subsection V-C to conduct the simulation with comparison of the trajectory privacy with different σ values. As seen in Figure 6, the trajectory privacy will be higher if the number of locations along a trajectory is larger. It is reasonable because if the trajectory is longer, the choices of dummy locations will be more. Similarly, if σ is higher, which means the voting scale is larger, there will be more choices of dummy locations. Consequently, the trajectory privacy will be preserved with larger σ value and longer trajectory.

In our paper, as described in Section III, the users are also able to communicate with the trustworthy party. We assume to encrypt users' true locations by traditional encryption algorithms like AES and 3DES. In this case, the security of transmitting information to trustworthy party is preserved by the hardness of compromising those encryption algorithms.

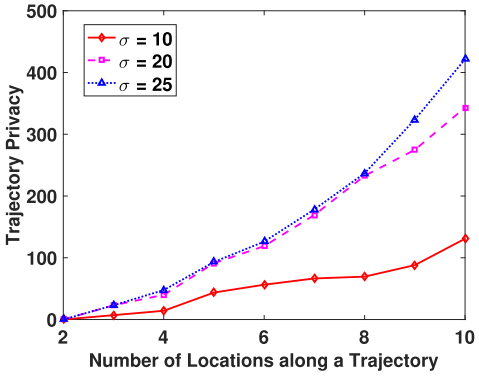


FIGURE 6. Trajectory privacy with different σ.

C. PERFORMANCE COMPARISON

The simulation setup is demonstrated in Subsection VII-A. In this subsection, we will discuss the result of energy cost minimization problem and compare the performance of the TPP scheme with random and rotation schemes [3].

As we mentioned in Section VI, we desire to transmit location information to the trustworthy party with lower energy consumption. Figure 7 indicates that as the number of locations along a trajectory becomes larger, more encrypted locations need to be transmitted to trustworthy parties for inferring the true trajectory. With a higher σ value, the encrypted locations is less. But when the number of locations are too small, no matter how big σ is, the performances are the same. Since we aim to show the tendency of the user's trajectory, σ value is not limited by the preferred walking speed of human.

Finally, we compare the defined trajectory privacy of our TPP scheme with the random and rotation schemes [3]. The number of generated random dummy locations is the same as locations along the true trajectory. The source and destination locations are the same along true and dummy trajectories. We randomly choose dummy locations to satisfy the walking speed of human. Since the source and destination locations are public known, we make some adjustments of the rotation

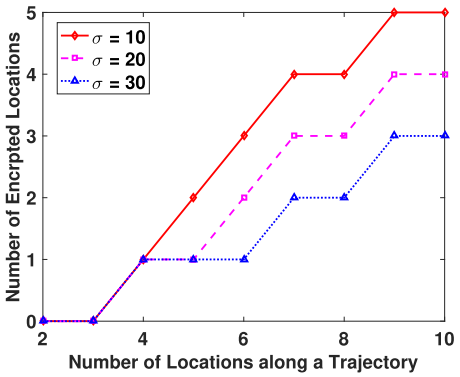


FIGURE 7. The number of encrypted location corresponding to different σ.

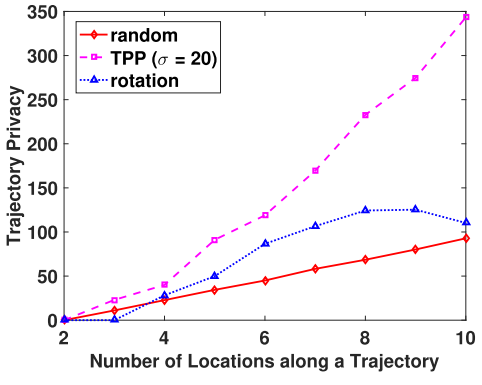


FIGURE 8. Performance comparison of different trajectory privacy preservation schemes.

scheme to fit this scenario. In the simulation, we only rotate the rest of locations 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 shown in Figure 8, we make a comparison between the random, rotation and our TPP scheme with σ equals to 20. It is obvious that our proposed scheme perform much better to preserve trajectory privacy.

VIII. CONCLUSION

In this paper, we have studied the trajectory privacy maximization problem via our proposed scheme against the tensor voting based inferring attack. We have introduced a novel trajectory inference attack model based on tensor voting theory. We have mathematically formulated the trajectory privacy maximization problem under several constraints such as saliency limitation based on tensor voting theory, and put it into an MILP problem. Because of the NP-hardness of the MILP problem, we have converted it into LP problem for an upper bound and developed a heuristic algorithm for feasible solutions. We also have illustrated the energy cost minimization problem. Through extensive simulations, we have shown that the proposed TPP scheme can effectively preserve LBS users' trajectory privacy.

REFERENCES

[1] G. Xu, S. Gao, M. Daneshmand, C. Wang, and Y. Liu, "A survey for mobility big data analytics for geolocation prediction," *IEEE Wireless Commun.*, vol. 24, no. 1, pp. 111–119, Feb. 2017.
[2] N. Ye, Y. Zhang, R. Wang, and R. Malekian, "Vehicle trajectory prediction based on hidden Markov model," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 7, pp. 3150–3170, Jul. 2016.
[3] 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.
[4] P.-R. Lei, W.-C. Peng, I.-J. Su, and C.-P. Chang, "Dummy-based schemes for protecting movement trajectories," *J. Inf. Sci. Eng.*, vol. 28, no. 2, pp. 335–350, 2012.
[5] T. Xu and Y. Cai, "Exploring historical location data for anonymity preservation in location-based services," in *Proc. 27th Conf. Comput. Commun. (INFOCOM)*, Phoenix, AZ, USA, Apr. 2008, pp. 547–555.
[6] P. Mordohai and G. Medioni, *Tensor Voting: A Perceptual Organization Approach to Computer Vision and Machine Learning*. Morgan & Claypool, 2006.
[7] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Proc. Int. Conf. Pervasive Comput.*, Munich, Germany, May 2005, pp. 152–170.
[8] T. Jiang, H. J. Wang, and Y.-C. Hu, "Preserving location privacy in wireless LANs," in *Proc. 5th Int. Conf. Mobile Syst., Appl. Services (MobiSys)*, San Juan, Puerto Rico, Jun. 2007, pp. 246–257.
[9] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Scottsdale, AZ, USA, Nov. 2014, pp. 251–262.
[10] 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, Jan. 2008.
[11] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. 1st Int. Conf. Mobile Syst., Appl. Services (MobiSys)*, San Francisco, CA, USA, May 2003, pp. 31–42.
[12] K. Vu, R. Zheng, and J. Gao, "Efficient algorithms for K-anonymous location privacy in participatory sensing," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Orlando, FL, USA, Mar. 2012, pp. 2399–2407.
[13] H. Lu, C. S. Jensen, 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 (MobiDE)*, Vancouver, BC, Canada, Jun. 2008, pp. 16–23.
[14] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing privacy through caching in location-based services," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Hong Kong, Apr./May 2015, pp. 1017–1025.
[15] F. Tang, J. Li, I. You, and M. Guo, "Long-term location privacy protection for location-based services in mobile cloud computing," *Soft Comput.*, vol. 20, no. 5, pp. 1735–1747, May 2016.
[16] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Berlin, Germany, Nov. 2013, pp. 901–914.
[17] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Denver, CO, USA, Oct. 2015, pp. 1298–1309.
[18] 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. (INFOCOM)*, Orlando, FL, USA, Mar. 2012, pp. 972–980.
[19] B. Palanisamy and L. Liu, "MobiMix: Protecting location privacy with mix-zones over road networks," in *Proc. IEEE 27th Int. Conf. Data Eng. (ICDE)*, Hannover, Germany, Apr. 2011, pp. 494–505.
[20] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Comput.*, vol. 2, no. 1, pp. 46–55, Jan. 2003.
[21] E. Pan, M. Pan, and Z. Han, "Tensor voting techniques and applications in mobile trace inference," *IEEE Access*, vol. 3, pp. 3000–3009, 2015.
[22] E. Pan, M. Pan, Z. Han, and V. Wright, "Mobile trace inference based on tensor voting," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Austin, TX, USA, Dec. 2014, pp. 4891–4897.
[23] Z. Han, M. Hong, and D. Wang, *Signal Processing and Networking for Big Data Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
[24] 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.
[25] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. Comput.*, vol. C-20, no. 1, pp. 68–86, Jan. 1971.
[26] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Series of Books in the Mathematical Sciences). New York, NY, USA: W. H. Freeman and Company, 1979.
[27] 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. (INFOCOM)*, Orlando, FL, USA, Mar. 2012, pp. 1557–1565.
[28] ILOG IBM. (2014). *Cplex Optimization Studio*. [Online]. Available: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer
[29] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*. Hoboken, NJ, USA: Wiley, 2011.
[30] C. DelBello, K. Raihan, and T. Zhang, "Reducing energy consumption of mobile phones during data transmission and encryption for wireless body area network applications," *Secur. Commun. Netw.*, vol. 8, no. 17, pp. 2973–2980, Nov. 2015.
[31] 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, Belgium, in 2016. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Houston. 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, China, in 2012, and the M.S. degree in electrical and computer engineering from Auburn University, Auburn, AL, USA, in 2015. She is currently pursuing the Ph.D. degree with 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 Best Paper Award in Globecom 2017.

MINGLEI SHU received the B.S. degree in automation, the M.S. degree in power electronics, and the Ph.D. degree in communication and information systems from Shandong University, China, in 2003, 2006, and 2017, respectively. He is currently a Research Fellow with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China.

He is also the Head of the Information Medicine Team, Shandong Computer Science Center (National Supercomputer Center in Jinan), the Executive Director of the Sino-Australian Joint Laboratory of International Health Technology, the Vice President of the Medical and Health Branch of Shandong Internet of Things Association, the Vice President of the "Internet +" Alliance, the Executive Director of the Telemedicine and Information Technology Branch of China Medical Equipment Association, and the Executive Director of "Internet +' Medical Professional Committee of China Health Information and Big Date Association. His research interests include medical artificial intelligence, medical big data and medical Internet of Things, wireless sensor networks, wireless body area networks, and information security.

YINGLONG WANG received the B.S. degree in electronic technology and the M.S. degree in industrial automation from the Shandong University of Technology, Jinan, China, in 1987 and 1990, respectively, and the Ph.D. degree in communication and information systems from Shandong University, Jinan, in 2005. He is currently a Research Fellow with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan. He is the Director of the Sino-Australian Joint Laboratory of International Health Technology, the Vice Chairman of the Shandong Science and Technology Association, the President of the Shandong Internet of Things Association, a member of the Shandong Information Expert Group, a member of the Shandong Information Expert Consultation Committee, the Vice Chairman of the Shandong Computer Society, and the Vice Chairman of the Shandong Information Standardization Technology Committee. His current research interests include medical artificial intelligence, high-performance computing, wireless sensor networks, information security, and cloud computing.

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

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

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate at the University of Maryland, College Park. From 2006 to 2008, he was an Assistant Professor at Boise State University, Boise, ID, USA. He is currently a Professor with the Electrical and Computer Engineering Department and with the Computer Science Department, University of Houston, TX, 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, the 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.