Privacy-Preserving Compressive Sensing for Trajectory Recovery in Mobile Social Networks

Linghe Kong*†, Liang He*, Xiao-Yang Liu†, Yu Gu*, Min-You Wu†
*Singapore University of Technology and Design, Singapore
†Shanghai Jiao Tong University, China
*{linghe_kong, he_liang, jasongu}@sutd.edu.sg, †{linghe.kong, yanglet, mwu}@sjtu.edu.cn

***Abstract*—Location based services (LBS) have experienced an explosive growth recently and evolved from utilizing a single user location to the whole trajectory of the user. However, due to the hardware and energy constraints, there are usually many missing data within a trajectory. In addition, as the trajectory information exposes the daily activities of a user, the privacy issue has been a major concern from mobile users. While there are effective state-of-the-art solutions that independently tackle the trajectory recovery and privacy preservation of user trajectories, yet no single design is able to tackle these two challenges simultaneously. Therefore in this paper, we propose a novel *Privacy Preserving Compressive Sensing* (PPCS) scheme to achieve both privacy preservation and accurate trajectory recovery. An encryption method named K-*Vector Perturbation* (KVP) is the major component of PPCS, which is proposed to perturb user trajectory with K other trajectories while maintaining the homomorphic encryption property for compressive sensing-based trajectory recovery. With PPCS, adversaries are only able to capture the encrypted data, and thus guarantees the privacy of mobile users. Furthermore, the homomorphic property guarantees that the recovery accuracy is comparable to state-of-the-art compressive sensing designs. The performance of PPCS is extensively evaluated through trace-driven simulations, which are based on two publicly available mobility traces from Beijing and Shanghai with user mobility models including walk, bike, and car. The results demonstrate that our design is able to achieve a recovery accuracy of $< 53\ m$ and an average distortion (a commonly adopted metric to evaluate the privacy preservation strength) of more than $9,000\ m$ even when 50% original data are missing.**

I. Introduction

Location based services (LBS) [17] [24] [30] in mobile networks have experienced an explosive growth recently, which are evolving from utilizing a single location [8] to harness the complete trajectory of a mobile user [39] for applications such as trajectory based forwarding [23], participatory sensing [5], and preference analysis [37].

While GPS is universally available on modern mobile devices, the trajectory of a mobile user may not be always complete due to none-line-of-sight to satellites [25]. In addition, since GPS consumes a significant amount of energy, it is normally activated periodically to conserve the energy on mobile devices [21]. Consequently, the trajectory recovery [26] is one of the fundamental components of LBS to reconstruct the trajectory of a mobile user when some of her locations data are missing. For example, trippermap [3] in Flickr is able to automatically reproduces a user 's travelling path based on her geotagged photos.

Considerable amount of effective interpolation methods have been devoted to trajectory recovery. With a single user's location data, methods such as nearest neighbors [27], linear [29], and Lagrange [35] methods have achieved reasonable accuracy. More recently, Rallapalli et al. [25] reveal that the trajectories of mobile users are normally strongly correlated within a geographic location. For example, students on a university campus have similar time tables; vehicles on the same segment of freeway move with the similar velocities. Leveraging on such correlations, they have proposed a social recovery method that collectively recovers all users' trajectories together using compressive sensing (CS), which proved to be superior to methods with only single user data [25]. While achieving better accuracy, the major drawback of such CS-based social recovery method is that it requires users to transmit their location data to a central server, which poses great concerns for potential privacy leakage.

On the other hand, privacy is drawing increasing attentions in social networks [38]. Even several well-known social networks have been reported privacy leakage events. For example, it is reported on Dec. 5, 2013 that two million logins and passwords from Facebook, Google and Twitter are stolen by botnet 'Pony'. Not only the logins but also the trajectories include users' sensitive information, and thus many works study the trajectory privacy in social networks. The most commonly adopted approach is anonymization [22]. However, latest studies [12] [32] have revealed that the anonymization-only solutions are inadequate in practice. To further improve the privacy, dummification [18] methods and obfuscation [13] [15] methods are introduced, which inject false trajectories and perturb original trajectories, respectively. Although dummification and obfuscation methods reasonably protect user privacies, they also pollute the original sensed data, which decrease the accuracy of trajectory recoveries with current social recovery methods.

To tackle the challenge of achieving accurate trajectory recovery while preserving the user privacy, we design a novel encryption method named K-vector perturbation (KVP) to attain both objectives. The main idea of KVP is to perturb a user's trajectory with K other trajectories while maintaining the homomorphic encryption property [10] [31] for compressive sensing. Based on KVP, we propose a privacy-preserving compressive sensing (PPCS) design that includes three major

steps. First, every user encrypts her original data by KVP and transmits the encrypted data to the server. Second, the server recovers all users' encrypted data with CS. Third, each user downloads the corresponding recovered data and decrypts her own trajectory by inverse KVP. Under PPCS, any adversary is only able to capture the encrypted data, which guarantees the privacy of mobile users. Furthermore, the homomorphic encryption property also guarantees the recovery accuracy even the CS-based recovery is operated on the encrypted data.

The major contributions of this paper are summarized as follows:

- To the best of our knowledge, this is the first work to jointly optimize the data recovery accuracy and user privacy preservation in mobile social networks.
- We propose a systematic PPCS design for social trajectories recovery, which combines the novel homomorphic encryption method KVP into compressive sensing framework to achieve the recovery accuracy and the user privacy preservation simultaneously.
- We theoretically prove that the recovery errors of PPCS and CS are within the same bound. We also derive that the expectation of distortion between the encrypted data and the original data is relatively large compared to the size of the area, which indicates effective data perturbation for privacy-preservation. Moreover, we prove that PPCS can protect the user privacy as long as there are no more than K location data of a user being exposed, where K is the length of the encryption key utilized in PPCS and can be proactively controlled according to user requirement on privacy preservation.
- Extensive trace-driven simulations are conducted to evaluate PPCS, which are based on two publicly available mobility traces from Beijing and Shanghai with diverse data scales, sizes of areas, and mixed mobility modes including walking, biking, driving and so on. The evaluation results show the effectiveness of PPCS. Typically, using PPCS on Beijing traces achieves the average accuracy within 53 meters and the average distortion more than $9,000$ meters even 50% original data are missing.

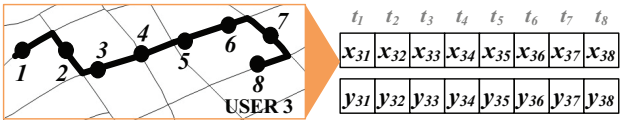
The remainder of this paper is organized as follow. In Section II, we formulate the trajectory recovery problem. In Section III, we investigate the mobility property in real traces. We describe our solution in Section IV, and analyze the theoretical bounds of recovery accuracy and privacy in Section V. In Section VI, we evaluate our solution based on trace-driven simulation. In Section VII, we review the related work. And we conclude in Section VIII.

II. Preliminaries

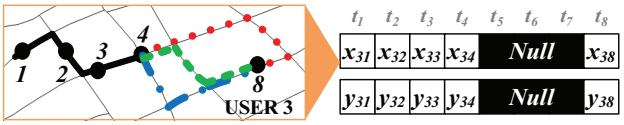
In this section, we describe the trajectory recovery model, the adversary model, and the formal definition of our problem.

A. Trajectory Recovery Model

A trajectory is composed of a set of locations that a user traverses, represented by the corresponding longitude x and the latitude y, as shown in Fig. 1(a). The user current location



(a) Two vectors are used to record the longitude x and the latitude y data of user 3's trajectory.



(b) When some location data are missing, the corresponding elements in the vectors are null. It is not easy to directly recover the accurate trajectory due to several possible paths.

Fig. 1. Trajectory model.

(x, y) can be obtained through the GPS module on her mobile device. In an N-user system where the total duration of interests consists of T time slots, the trajectory of each user is represented by two $1 \times T$ vectors, where x_{ij} and y_{ij} are the longitude and latitude at the j-th time slot respectively ($i = 1, 2, \cdots, N$ and $j = 1, 2 \cdots, T$).

In practice, the location data of a user could be partially missing due to reasons such as none-line-of-sight to GPS satellites, energy management of GPS on mobile devices [21] and so on. In Fig. 1(b), the missing location data are represented by the *null* element in the vectors.

The trajectory recovery is not effective if it is performed for individual users independently. For example, as shown in Fig. 1(b), when the location data for the 5-th to the 7-th time slots are missing, we can get three possible trajectories if linear interpolation [27] and map matching [28] are utilized to recover the trajectory. To address the weakness of the single user recovery, the CS-based social recovery exploits the correlation among users and collectively recovers N users' trajectories simultaneously, which is shown to outperform most existing methods [25] and is treated as the state-of-the-art in this paper.

A few notations related to social recovery are defined as follows.

- *Trajectory Matrix* is a set of N users' actual trajectories, which is defined as $X = (x_{ij})_{N \times T}$. We only illustrate the longitude X related definitions and derivations in the following sections. All results for the latitude Y are identical to X, which are omitted for conciseness.
- *Binary Index Matrix* is used to indicate whether a location data at the corresponding position in X is missing, which is defined as

$$\Phi = (\phi_{ij})_{N \times T} = \begin{cases} 0 & \text{if } x_{ij} \text{ is missing,} \\ 1 & \text{otherwise.} \end{cases} \quad (1)$$

- *Sensed Matrix* S consists of the sensed location data by GPS. Due to the potentially missed data, elements in S are either x_{ij} (i.e., the sensed location data) or 0 (i.e., indicating the missing data). Thus, S can be presented

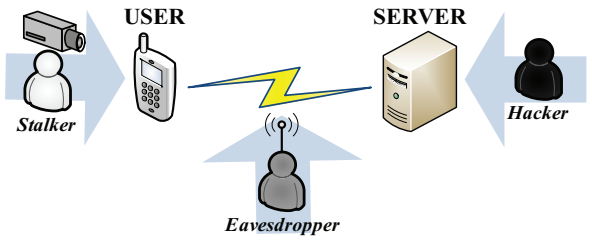


Fig. 2. Adversary models.

by[1]

$$S = X \circ \Phi. \tag{2}$$

- *Recovered Matrix* is generated by recovering the missing data in the sensory matrix S to approximate the actual trajectories X. The recovered matrix is denoted by $\hat{X}$.
- *Compressive Sensing* (CS) is a social recovery method to recover the missing data in S. We use f_{cs} to denote the CS-based recovery operation, and thus $\hat{X} = f_{cs}(S)$.

B. User Models and Adversary Models

We consider a system consisting of two types of mobile users: public and private users. Public users are willing to share their trajectories and private users want to protect the privacy of their trajectories. For example, in an urban traffic scenario, buses can be treated as public users, and personal vehicles are good examples of private users. In a university campus, the campus shuttles and students can be treated as public users and private users, respectively.

As leakage of personal trajectories can lead to unauthorized surveillance and tracking, adversaries are motivated to obtain private users' trajectory information. In Fig. 2, we illustrate the adversary models that threaten the trajectory privacy in social recovery, which are categorized as eavesdroppers, hackers, and stalkers.

• *Eavesdroppers (hackers)*: An eavesdropper could potentially capture any data transmitting between private users and the server due to the unsecured communication channels. A hacker could access and obtain all data in the server. Because eavesdroppers and hackers could potentially obtain the same set of information, we do not differentiate them in the rest of the paper.

• *Stalkers*: A stalker has the same ability of an eavesdropper. Moreover, a stalker is able to obtain k actual location data of a user when he successfully stalks or occasionally encounters this user.

The adversaries also potentially have the following capabilities: (i) they have access to the exposed data and the same social recovery algorithms as ours to estimate the trajectory; (ii) they can exploit map matching methods [28] and the other public information such as road speed limits to further improve their estimation accuracy.

[1]In this paper, $X\Phi$ presents the matrix production of X and Φ. And $X \circ \Phi$ presents the element-wise production of X and Φ.

TABLE I
SELECTED REAL-WORLD MOBILITY TRACES

Name	Size	Interval	Area	Mode
Beijing	116×355	5 sec	70×85 km^2	Walk/Bike/Car
Shanghai	74×399	1 min	100×100 km^2	Taxi/Bus

C. Problem Definition

In this paper, we consider the *accurate and privacy-preservation trajectory recovery* problem. On one hand, we desire that the highest accuracy can be achieved for every user's trajectory recovery. On the other hand, we desire that the attacks from eavesdroppers, hackers, and stalkers can be effectively defended.

The problem is challenging because the two objectives appear to be conflict with each other. In order to achieve the accuracy objective, social recovery requires to collect data from all users into a central server. In contrast, the privacy objective is to avoid the exposure of user data. Existing methods cannot satisfy the two objectives simultaneously. Therefore, a new design is desirable to address this dilemma in trajectory recovery.

In the proposed PPCS, the user privacy is preserved through a novel encryption method and CS is applied on the encrypted data to accurately recover the trajectories.

III. TRACE PREPROCESSING AND VALIDATION

Before introducing the design of PPCS, we first introduce two mobility traces studied in this paper and validate their low-rank properties, so the proposed CS operation can be effectively applied.

A. Preprocessing of Real-World Mobility Traces

The evaluation of our design is based on two publicly available mobility traces: Geolife [1] and SUVnet [2]. Geolife records the GPS trajectories of 178 users from April 2007 to October 2011 in Beijing, in which the major user mobility modes include walking, biking, and driving. The longitude and latitude data in this trace are accurate to 6 decimal places. SUVnet records the trajectories of over 2000 taxis and 300 buses in the urban area of Shanghai, whose data are accurate to 4 decimal places. However, the raw traces from Geolife and SUVnet cannot be directly utilized for low rank validation, because significant amount of there data are missing. To guarantee the existence of ground truth location information, we perform trace preprocessing on the raw traces to select their complete subsets and build the trajectory matrices, which are then utilized in our evaluations. The description of the two selected traces are shown in Table I, which are denoted as Beijing trace and Shanghai trace respectively.

B. Validating the Low Rank Property

As CS is also a major component of the proposed PPCS, we first need to validate whether the trajectory matrices demonstrate the low-rank property, which is required for the CS operation to achieve accurate trajectory recovery [34]. Note

that although the low-rank property of certain traces has been shown in [25], each of their traces has only one mobility mode: either human walking or car driving. The mobility mode mixed with walking, biking, and driving together in our selected traces is a more general scenario. Hence, we still need to verify whether such traces are universally low rank.

We verify the low-rank property of the selected traces through *Singular Value Decomposition* (SVD), which is an effective non-parametric technique for rank investigation [20].

According to SVD, a $N \times T$ matrix X can be decomposed as

$$X = U\Lambda V' = \sum_{i=1}^{min(N,T)} \sigma_i u_i v_i', \tag{3}$$

where U and V are two unitary matrices, V' is the transpose of V, and Λ is a $N \times T$ diagonal matrix containing the singular value σ_i of X.

Typically, the singular values σ_i are sorted in that $\sigma_i \geq \sigma_{i+1}, (i = 1, 2, \cdots, min(N,T))$, where $min(N,T)$ is the number of singular values. The rank of the matrix X, denoted by r, is the number of its non-zero singular values. The matrix is low-rank if $r << min(N,T)$. In practice, if the top-$\hat{r}$ singular values have a good approximation of the total singular values, i.e.,

$$\sum_{i=1}^{\hat{r}} \sigma_i \approx \sum_{i=1}^{min(N,T)} \sigma_i, \tag{4}$$

this matrix is considered to be near low-rank, and $\hat{r}$ is treated as its rank.

The CDF of the singular values obtained with the Beijing and Shanghai traces are shown in Fig. 3, where the x-axis presents the i-th largest singular values, and the y-axis is the ratio between the sum of the top-i singular values and the sum of all singular values. We can see that the total singular values can be well approximated by only a few top singular values in both traces. For example, the top-7 σ_i of the Beijing trace and the top-13 σ_i of the Shanghai trace occupy more than 95% of their respective total values, while the total numbers of σ_is are 116 and 74 respectively. This observation reveals that the two traces investigated in our work is of the near low-rank property, and thus the CS operation can be applied to achieve promising recovery accuracy.

Note that with the near low-rank property in CS, $1 - (\sum_{i=1}^{r} \sigma_i / \sum_{i=1}^{min(N,T)} \sigma_i)$ can be considered as the noise [6], which negatively affects the recovery accuracy of CS.

IV. PPCS Design

To address the potential privacy exposure issue with the CS-based approach, we present a simple but efficient trajectory recovery method *Privacy-Preserving Compressive Sensing* (P-PCS), which improves the user privacy protection while guaranteeing promising trajectory recovery accuracy. We introduce the PPCS scheme in this section, and the detailed analysis on its performance will be presented in Section V.

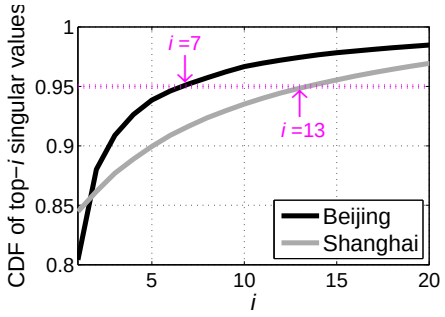


Fig. 3. Low-rank property in the investigated mobility traces.

A. Design Overview

The proposed PPCS consists of three steps. In the first step, users encrypt their sensed data and transmit the encrypted trajectories to the server. Note these encrypted trajectories may not be complete because of the data missing issue. The server then performs CS on the collective data received from users to recover the missing part of the encrypted trajectory for all users. At last, individual users downloads their recovered and encrypted trajectory from the server and decrypts it to obtain her original trajectory. An overview of these three steps is shown in Fig. 4.

B. Encrypt the Sampled Trajectories at Individual Users

The core component of PPCS is to encrypt the sensed trajectories at private users, and thus only their encrypted trajectories are available at the server. Denote f_{en} as the encryption operation, with a sensed trajectory $S_{(i)}$ of the user i, the encrypted trajectory can be represented as

$$\mathbb{S}_{(i)} = f_{en}(S_{(i)}), \tag{5}$$

where $S_{(i)}$ presents the i-th row vector in the matrix S.

In the next, we explain how the encryption operates in detail. In the system under consideration, the public users are willing to share their trajectories with others, which are available at the server. At the first step of the encryption, a private user i randomly downloads K public vectors $D_{(1)}, D_{(2)}, \cdots, D_{(K)}$ from all public vectors in the server, which is utilized to generate the encrypted vector $\mathbb{S}_{(i)}$. Then user i generates a length-$(K+1)$ random vector $<\psi_{i,0}, \psi_{i,1}, \psi_{i,2} \cdots, \psi_{i,K}>$ as her *key* for the encryption, which is known only to herself. Any key satisfies $\psi_{i,j} \in (0,1)$ and $\sum_{j=0}^{K} \psi_{i,j} = 1$. With the public vectors and the generated encryption key, user i generates the encrypted vector $\mathbb{S}_{(i)}$ according to

$$\mathbb{S}_{(i)} = (\psi_{i,0} S_{(i)} + \psi_{i,1} D_{(1)} + \cdots + \psi_{i,K} D_{(K)}) \circ \Phi_{(i)}, \tag{6}$$

where $\circ$ represents the element-wise production.

To demonstrate the encryption operation, let us consider the example shown in Fig. 5. Assume a private user $i = 4$ has downloaded $K = 2$ public vectors from the server (*i.e.*, $D_{(1)}, D_{(2)}$), and has generated the length-3 key $<\psi_{4,0}, \psi_{4,1}, \psi_{4,2}>$. The three vectors $S_{(4)}$, $D_{(1)}$, and $D_{(2)}$ are summed up with weight $\psi_{4,0}$, $\psi_{4,1}$, and $\psi_{4,2}$ respectively.

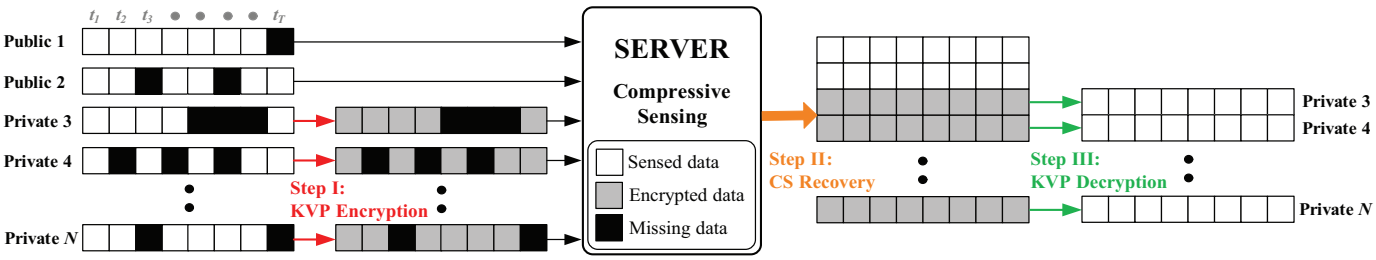


Fig. 4. PPCS overview.

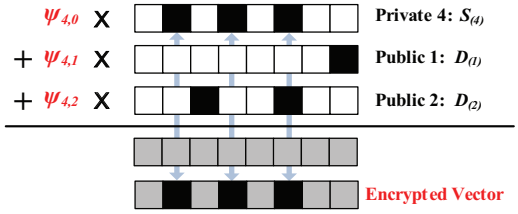


Fig. 5. KVP encryption

For each null element in $S_{(4)}$, the corresponding element in the resultant sum vector is treated as the missing data and the encrypted vector $\mathbb{S}_{(4)}$ is then transmitted to the server.

This encryption method is referred to as K-Vector Perturbation (KVP) in this paper, because 1) from the aspect of matrix operation, f_{en} is essentially a linear combination of K *vectors*in a matrix [16], and 2) the physical meaning of f_{en} is to *perturb* the user trajectory with the K public trajectories.

Intuitively, the length of encryption key dominates the difficulty for adversaries to decrypting the original data, and thus the value of K determines the privacy protection strength offered by KVP. We will further discuss the impact of K on the performance of PPCS in Section V-C.

C. Recover the Encrypted Trajectories at the Server

After collecting the encrypted trajectories from all private users and original trajectories of all public users, each consisting of T time slots, the server forms a encrypted matrix $\mathbb{S}$ of size $N \times T$. Then compressive sensing is applied on $\mathbb{S}$ and the completed encrypted trajectory matrix is obtained as $\hat{\mathbb{X}} = f_{cs}(\mathbb{S})$.

The detailed CS operation [6] is out the scope of this paper, and thus is not included here for the space limitation.

D. Decrypting the Recovered Trajectories at Individual Users

After the encrypted trajectories are recovered at the server, individual users can download their corresponding encrypted trajectories and apply the decryption operation. Specifically, user i downloads the recovered trajectory $\hat{\mathbb{X}}_{(i)}$ from the server and decrypts it with the public vectors and the local keys utilized in the encryption phase

$$\hat{X}_{(i)} = (\hat{\mathbb{X}}_{(i)} - (\psi_{i,1}D_{(1)} + \cdots + \psi_{i,K}D_{(K)}))/\psi_{i,0}. \quad (7)$$

V. PPCS Analysis

In this section, we analyze the performance of PPCS in three metrics: the trajectory recovery accuracy, the privacy protection against eavesdroppers, and the privacy protection against stalkers. The complexity analysis of PPCS is also presented.

A. Accuracy Analysis

Although CS-based trajectory recovery method has been shown to achieve promising accuracy [25], we need to make sure the encryption operation does not degrades the trajectory recovery accuracy.

We adopt the same metric in [25] to evaluate the recovery accuracy, namely, the *recovery error* ϵ. For user i, its recovery error $\epsilon_{(i)}$ is the geometric mean of the distance between the actual trajectory and the recovered trajectory, defined as

$$\epsilon_{(i)} = \frac{||X_{(i)} - \hat{X}_{(i)}||}{T}, \quad (8)$$

where $||X_{(i)} - \hat{X}_{(i)}|| = \sqrt{\sum_{j=1}^{T}(x_{ij} - \hat{x}_{ij})^2}$, and T is the total number of time slots along the trajectory.

With this accuracy metric, we have the following theorem stating that the encryption operation KVP does not degrades the data recovery accuracy.

Theorem 5.1: The proposed KVP is a homomorphic encryption method for CS. That is, if a matrix X is near low-rank, the recovery accuracy of a user i satisfies

$$\sup ||X_{(i)} - \hat{X}_{(i)}|| = \sup ||X_{(i)} - \tilde{X}_{(i)}||, \quad (9)$$

where sup is upper bound of $||\cdot||$, $\hat{X}$ is the trajectories recovered by CS with KVP (i.e., $\hat{X} = f_{de}(f_{cs}(f_{en}(X \circ \Phi)))$), $\tilde{X}$ is trajectories recovered by CS directly (i.e., $\tilde{X} = f_{cs}(X \circ \Phi)$), and $\hat{X}_{(i)}$ is the recovered trajectory of user i.

Proof: As mentioned in Section III, when a matrix is near low-rank and the value of approximate rank is r, the value $\sum_{i=1}^{min(N,T)} \sigma_i - \sum_{i=1}^{r} \sigma_i$ can be treated as noise, which is denoted as δ.

According to existing work on the CS-based matrix completion [6] [14], we have the accuracy upper bound as

$$\sup ||X - \tilde{X}|| = 4\sqrt{\frac{2\min(N,T)}{(1-\alpha)}}\delta_1, \quad (10)$$

where α is the data loss ratio in X, and δ_1 is the noise of X.

Similarly, the accuracy upper bound of $||\mathbb{X}||$ can be represented as

$$\sup||\mathbb{X}-\hat{\mathbb{X}}|| = 4\sqrt{\frac{2\min(N,T)}{(1-\alpha)}}\delta_2. \tag{11}$$

where δ_2 is the noise of $\mathbb{X}$.

From Fig. 5, we know that the KVP operation does not change the number of missing data. And thus the loss ratio α in Eq. (10) and in Eq. (11) has the same value. Combining Eq. (10) and Eq. (11), we have

$$\frac{\sup||\mathbb{X}-\hat{\mathbb{X}}||}{\sup||X-\tilde{X}||} = \frac{\delta_2}{\delta_1}. \tag{12}$$

It is difficult to obtain the exact value of δ_1 and δ_2, which depends highly on the specific data under consideration. However, because KVP is a basic linear transformation, which can be presented as $\mathbb{X} = \Psi X$ and Ψ is the matrix of keys ψ. Treating this transformation as a measurement operation in CS, we can obtain the noise ratio according to CS theory [6],

$$\frac{\delta_2}{\delta_1} = \frac{|\mu(\Phi,\Psi)|}{|\mu(\Phi,I)|}. \tag{13}$$

whre Φ is the 0-1 matrix indicating the missing data. The coherence operation μ in Eq. (13) is defined as

$$\mu(\Phi,I) = \max_{1\le i\ne j\le T} |<\Phi^{(i)}, I^{(j)}>|. \tag{14}$$

where $\Phi^{(i)}$ is the i-th column vector of $\Phi_{N\times T}$, and $<\Phi^{(i)}, I^{(j)}>$ is the inner product of two vectors, i.e., $<\Phi^{(i)}, I^{(j)}> = (\Phi^{(i)})' I^{(j)}$.

By definition, we have the Ψ matrix as follows, which is an example when $K=2$ as shown in Fig. 5

$$\Psi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \psi_{3,1} & \psi_{3,2} & \psi_{3,0} & 0 & 0 & 0 & 0 \\ \vdots & \vdots & 0 & \ddots & 0 & 0 & 0 \\ \psi_{i,1} & \psi_{i,2} & 0 & 0 & \psi_{i,0} & 0 & 0 \\ \vdots & \vdots & 0 & 0 & 0 & \ddots & 0 \\ \psi_{N,1} & \psi_{N,2} & 0 & 0 & 0 & 0 & \psi_{N,0} \end{bmatrix}. \tag{15}$$

Combine Eq. (12), Eq. (13), Eq. (14), and Eq. (15), we can calculate the recovery error of user i as

$$\frac{\sup||\mathbb{X}_{(i)}-\hat{\mathbb{X}}_{(i)}||}{\sup||X_{(i)}-\tilde{X}_{(i)}||} = \frac{|\mu(\Phi,\Psi^{(i)})|}{|\mu(\Phi,I^{(i)})|} = \frac{\psi_{i,0}}{1}. \tag{16}$$

Because of the reasons that 1) the decryption operation $f_{de}(\mathbb{X}_{(i)}) = \hat{X}_{(i)}$ is also a linear transformation (Eq. 7), 2) all other variables such as Ds and ψs are known, and 3) $\sum_{j=0}^{K}\psi_{i,j} = 1$, we know the error is linearly amplified according to weights

$$\frac{\sup||X_{(i)}-\hat{X}_{(i)}||}{\sup||\mathbb{X}_{(i)}-\hat{\mathbb{X}}_{(i)}||} = \frac{\psi_{i,0}+\psi_{i,1}+\cdots+\psi_{i,p}}{\psi_{i,0}} = \frac{1}{\psi_{i,0}}. \tag{17}$$

Combining the above two equations, we have

$$\sup||X_{(i)}-\hat{X}_{(i)}|| = \sup||X_{(i)}-\tilde{X}_{(i)}||, \tag{18}$$

and the theorem is proved. ■

B. Privacy Protection against Eavesdroppers

The motivation of PPCS is to offer privacy protection while guaranteeing recovery accuracy. We discuss how PPCS can protect privacy leakage against eavesdroppers (in this subsection) and stalkers (in the next subsection).

The sensed data are encrypted by individual users before aggregated at the server. In this way, only encrypted data (either the encrypted sensed trajectories sent from the users, or the complete encrypted trajectories recovered with CS) are available at the server. After the encryption, eavesdroppers can only infer the original user trajectory based on the exposed encrypted data $\hat{\mathbb{X}}$. Therefore, we adopt the *distortion* δ defined in [39] to measure the similarity between the encrypted and the original data of every user

$$\delta_{(i)} = \frac{\sum_{j=1}^{T}|\hat{\mathbb{X}}_{(i,j)}-\hat{X}_{(i,j)}|}{T}. \tag{19}$$

The value of δ indicates the average per-location distortion between the encrypted and the original trajectory, and a larger δ indicates a stronger privacy protection against eavesdroppers. Since $\mathbb{S}$ and S are incomplete, we use recovered $\hat{\mathbb{X}}$ and $\hat{X}$ to replace them for computing.

The PPCS design exploits KVP to obfuscate the user's personal trajectory. Since several trajectories are perturbed into one trajectory, even a eavesdropper steals this combined trajectory, it is not easy to distinguish the original one. In the next, we derive the distribution of the distortion δ.

The encrypted vector is obtained through linearly combining K public vectors with corresponding weights ψ, and this encryption operation demonstrates significant randomness in that 1) the K public vectors are randomly selected from all public vectors and 2) the weight vector $<\psi_{i,0},\psi_{i,1},\cdots,\psi_{i,K}>$ is randomly generated. With these randomness, the original locations are mapped to another random locations in the area of interests. As a result, we can use the random distance distribution in the area to approximate the distortion of a given location and its encrypted data.

Consider a $w\times h$ rectangle area, the distortion distribution $\mathbb{P}(\delta\le d)$ can be presented by a piecewise function [4]

$$\mathbb{P}(\delta\le d) = \begin{cases} \frac{2}{w^2h^2}(G(d)-G(0)) & d\in[0,h] \\ \frac{2}{w^2h^2}(G(h)-G(0)) & d\in(h,w] \\ \frac{2}{w^2h^2}(G(h)-G(\sqrt{d^2-w^2})) & \\ +F_h(\sqrt{d^2-w^2}) & d\in(d,\eta] \end{cases}, \tag{20}$$

where

$$\begin{aligned} G(z) &= \int(h-z)\sqrt{d^2-z^2}(2w-\sqrt{d^2-z^2})\mathrm{d}z, \\ F_h(z) &= 1-(1-z/h)^2, \end{aligned}$$

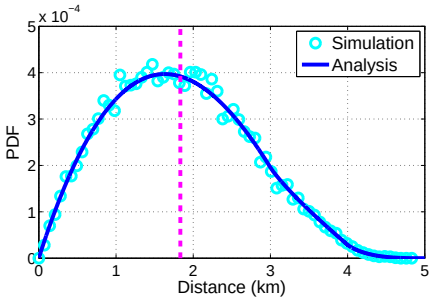


Fig. 6. PDF of the distortion by KVP.

and

$$\eta = \sqrt{w^2 + h^2}.$$

With this distribution, the average distance between randomly selected points (i.e., the expectation of distortion $\bar{\delta}$) can be easily obtained.

To valid our reasoning on the distortion distribution, we simulate a rectangle area with $w = 4\ km$ and $h = 3\ km$, and randomly generate a set of trajectories including a total number of 10^6 locations. Then we apply KVP on these trajectories and record the distances between the original locations and their corresponding encrypted locations. The statistic distribution of these distances is shown in Fig. 6, along with the probability distribution calculated according to Eq. (20). The average distance of these location pairs is also shown in the figure (i.e., $\bar{\delta} \approx 1.83\ km$).

Two observations can be obtained from these results. First, the distortion shows significant randomness over a large distance range, and second, the average distortion between the original and encrypted locations is relatively large in the area of interest. These two observations verify that there is no obvious pattern to infer the original locations through the encrypted locations.

C. Privacy Protection against Stalkers

The other adversary model we consider is the stalkers, who are more powerful than eavesdroppers in that they can obtain part of the original user trajectories. To protect the user privacy against stalkers, it is required that

$$\begin{array}{ll} \text{Given:} & \hat{\mathbb{X}}_{(i)}, f_{de}, \text{and } k \text{ location data in } X_{(i)}, \\ \text{Objective:} & \hat{X}_{(i)} \text{ is unsolvable.} \end{array} \tag{21}$$

The PPCS solution exploits the personal *keys* to protect privacy against the attacks from stalkers.

From Eq. (7), we know

$$\hat{\mathbb{X}}_{(i)} = \psi_{i,0}\hat{X}_{(i)} + \psi_{i,1}D_{(1)} + \cdots + \psi_{i,K}D_{(K)}. \tag{22}$$

It is possible for the stalker to obtain $\hat{\mathbb{X}}_{(i)}$ by hacking the server, and he may also obtain the public vectors D_i $(i = 1, 2, \cdots, K)$. However, because the encryption keys $\psi_{i,j}$ $(j = 0, 1, \cdots, K)$ are only known to the users themselves, for the stalker to resolve Eq. (22), he needs the knowledge of at least $K + 1$ elements of $X_{(i)}$ according to the theory of underdetermined system [9]. As a result, the stalker cannot resolve the original trajectory as long as the condition $k \leq K$ holds. As K is the control parameter adopted in PPCS, we can proactively adjust the number of public vectors used in the encryption operation to provide the privacy protection against stalkers according to the requirement of individual users.

D. Complexity Analysis

1) Computation Complexity: For the KVP operation, $K + 1$ vectors of size $1 \times T$ need to be processed, requiring a computation complexity of $\mathcal{O}((K + 1)T)$. This complexity is well within the power of current mobile devices with the processing capability of a few GHz.

On the server side, the main computation task is to recover the complete decrypted trajectories with CS, which requires a computation complexity of $\mathcal{O}(rNT\varrho)$ [19], where r is the rank of the to-be-recovered matrix and ϱ is the iteration numbers. Our evaluation experiences with the Beijing and Shanghai traces reveal that $\varrho \leq 5$ in most cases.

2) Communication Complexity: In order to execute f_{en}, a user should download K public vectors $D_{(i)}$ from the server and then upload a encrypted vector $\mathbb{S}_{(i)}$. Hence, the communication complexity is $\mathcal{O}((K + 1)T)$. On the other hand, in order to execute f_{de}, the user should download $\hat{\mathbb{X}}_{(i)}$, requiring another communication complexity of $\mathcal{O}(T)$. As an example, with $K = 10$, $T = 500$ and a 16-bit operating system, the total data exchange amount is about $(10 + 2) \times 500 \times 16/8 = 12\ KB$, which is well manageable for modern mobile applications.

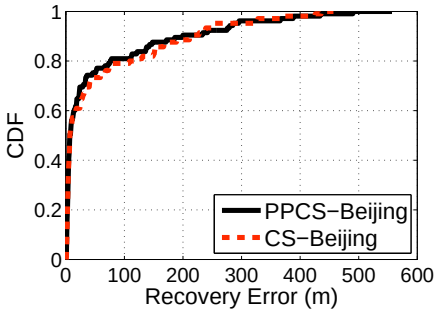
VI. Performance Evaluation

In this section, we evaluate the performance of PPCS in terms of both the data accuracy and privacy.

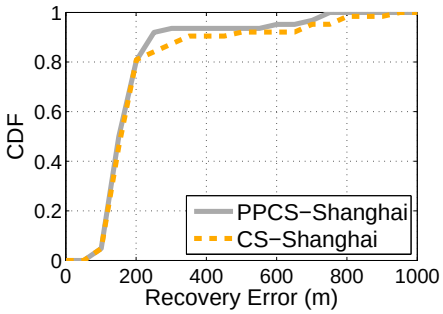
A. Simulation Settings

We evaluate PPCS based on two real-world mobility traces including walk, bike, and car traces in Geolife [1], and taxi and bus traces in SUVnet [2]. Using the same method as in Section III, we pre-process the raw data in Geolife and SUVnet by selecting small but complete trajectories and adopt these processed trajectories to carry out our simulation. The selected traces are named Beijing traces with the size of 116×355 and Shanghai traces with the size of 74×399, whose detailed descriptions are listed in Table I.

In the trace-driven simulation, we randomly generate a 0-1 matrix Φ with the same size as the original data trace. The elements in Φ take the value of 0 if the corresponding element in the data trace is missing and 1 otherwise. The ratio of the 0 elements to the number of all elements in Φ is controlled by the data loss ratio α, which is 0.5 unless otherwise specified. Then we generate the sensed matrix S according to Eq. (2) $S = X \circ \Phi$. The proposed PPCS is applied on the sensed matrix S with K public vectors, and the recovered matrix $\hat{X}$ is obtained. Without loss of generality, the top-K rows in the original traces are treated as the public traces, and $K = 10$

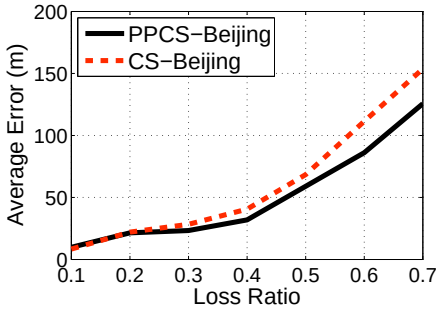


(a)

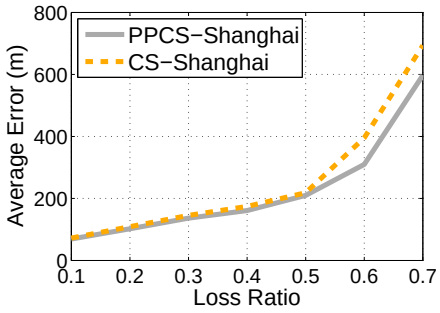


(b)

Fig. 7. Recovery accuracy distribution.

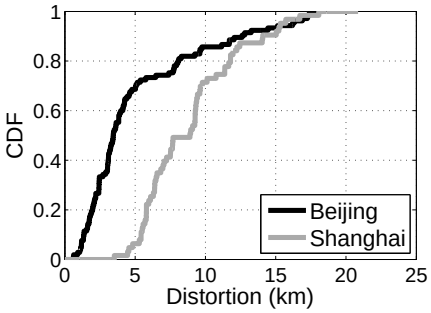


(a)

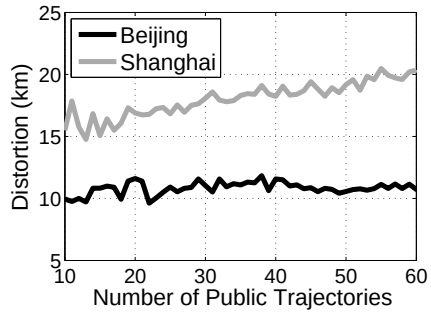


(b)

Fig. 8. Recovery accuracy vs loss ratio.



(a) Distortion distribution



(b) Distortion vs K

Fig. 9. Distortion performance.

by default. The reported results in the following are averaged over 100 simulation runs.

As a baseline, we adopt the state-of-the-art CS-based trajectory recovery method proposed in [25] to compare with the proposed PPCS, which is referred to as CS for short in the remaining of this section.

B. Performance Analysis

1) Recovery Accuracy: We first evaluate the recovery accuracy with a default setting of $\alpha = 0.5$ and $K = 10$. Figure 7 shows the distributions of the recovery accuracy obtained by PPCS and CS with the two real-world traces. For example, Fig. 7(a) shows the recovery errors of 50% users' trajectories are less than 10 meters when applying PPCS on the Beijing trace. Two observations can be obtained from this set of figures. First, the recovery accuracy obtained with PPCS and CS are comparable in both traces, which validates the correctness of Theorem 5.1 that PPCS can achieve similar recovery accuracy as the state-of-the-art CS method. Second, the recovery errors are small. For instance, the recovery errors of 80% users with the Beijing and Shanghai are less than 100 m and 200 m, respectively. These errors are tolerable in many cases because mechanisms such as map matching [28] can be adopted to eliminate their impacts on the final recovered trajectories.

To gain more insights on the impact of data loss on recovery accuracy, we apply PPCS on the two traces with varying α from 0.1 to 0.7, and the results are shown in Fig. 9. A clear increasing trend of recovery errors with the increase of α can be observed, which is intuitive. For example, with the Beijing trace, PPCS achieves an average recovery error of 20 m when $\alpha = 0.2$, and the error is increased to 124 m with an α of 0.7. The recovery errors with PPCS and CS are comparable in all the explored cases, which agrees with the observation in Fig. 7. An interesting observation is that PPCS slightly improves the accuracy performance of CS when data loss ratio grows. A possible reason is that the loss in noise also increases as more data is missing. As a result, the linear transformations in KVP make the low-rank property of the trajectory matrix even more obvious, and thus improves the recovery accuracy.

2) Privacy against Eavesdroppers: Keeping $\alpha = 0.5$ and $K = 10$, next we investigate the perturbation distortion obtained with PPCS, and the results are shown in Fig. 9(a). We can see that the distortion between the original and encrypted trajectories is enormous. For example, the distortion of 50% trajectories are over 4000 m with the Beijing trace. This distortion distance is quite large when compared with the road segment length in Beijing City. As a result, even if the encrypted trajectory is exposed to adversaries, the information leakage on the original trajectory is small, indicating a strong privacy protection level. Another observation from these results is that the distortion distribution shows no clear patterns. For example, the distortion distribution with the Shanghai trace is nearly linear but that for the Beijing trace more like a piecewise function. These patternless feature of the distortion distribution indicates that even the adversaries can obtain many of the encrypted trajectories, the training based on these information would not facilitate them to infer the original trajectories.

PPCS needs a number of public trajectories to perform the

(a) (b) (c)

Fig. 10. Illustrative results of PPCS.

TABLE II
RECOVERY ERROR WHEN STALKERS CAN OBTAIN 5% AND 10% OF THE ORIGINAL TRAJECTORIES.

Traces	Stalker (5%)	Stalker (10%)	User (50%)
Beijing	409.36	366.47	38.99
Shanghai	2510.48	1723.16	189.96

encryption. To investigate the impact of the amount of public trajectories on the distortion, we apply PPCS on the two traces with the number of public traces varying from 10 to 60 (the total number of users in Beijing and Shanghai are 116 and 74 respectively). The resultant distortion demonstrate no clear relationship with how many public traces are available. This observation alleviates our concern on whether the available number of public trajectories will significantly degrades the distortion performance of PPCS.

3) Privacy against Stalkers: Besides eavesdropping the encrypted trajectories, it is also possible for adversaries to directly obtain part of the original trajectories, i.e., the stalkers. The stalker can treat the exposed k data of a user as a $(T-k)$ missing data trajectory and then also utilizes PPCS to recover this trajectory. In the next, we evaluate the privacy protection offered by PPCS against such adversaries, and the results are shown in Table II. When a stalker has 5% original trajectories, the recovery error is more than $2,500$ m with the Shanghai trace and more than 400 m with the Beijing trace, which is difficult to obtain the correct trajectory with such a large error. Even a stronger stalker with 10% real trajectories cannot achieve promising recovery accuracy when compared with the users with a data loss ratio of $\alpha = 0.5$. On the contrary, the 50% private user has an excellent result that always under $40m$ with Beijing trace. In summary, our PPCS solution can protect the privacy even a few original data are exposed.

C. Illustrative Results

To demonstrate a clear view of the results obtained with PPCS, we show the recovered trajectory, the encrypted trajectory, and the stalker recovered trajectory in Fig. 10, with a 10-location original trajectory.

Figure 10(a) shows the recovered trajectories by PPCS and the CS method proposed in [25] when 4 locations along the trajectory are missing from the sensed trajectory. We can see even 40% of original data are missing, PPCS can still recover the original trajectory with a high accuracy that is comparable to the results achieved by directly utilizing CS.

The distortion between the original and encrypted trajectories (after processed with the map matching method proposed in [28]) are shown in Fig. 10(b). We can see that first, the distance between the two trajectories are relatively large, indicating a strong defense against eavesdroppers. Furthermore, the obtained encrypted results are still a sound trajectory according to the map. This indicates that the eavesdroppers cannot easily determine whether the hacked trajectories are encrypted or not.

Figure 10(c) shows the recovered trajectory of a stalker who applies PPCS with 3 obtained original trajectory locations, which is a totally different trajectory when compared with the ground truth.

VII. RELATED WORK

Two important research topics are involved in this work: trajectory recovery and trajectory privacy, which we literate briefly in this section.

A. Trajectory recovery

Existing research efforts on trajectory recovery can be classified into two categories: *single user recovery* and *social recovery*.

The single user recovery is to reconstruct a user's trajectory based on her own sensed location data. Many classic missing data estimation methods such as nearest neighbors (NN) [29], linear interpolation [27], and Lagrange interpolation [35] can be utilized to recover the user trajectory in mobile devices. These methods avoid the data leakage issue because no data exchange is required; however, the achieved trajectory recovery accuracy is usually limited [25].

The social recovery is to reconstruct all users' trajectories together based on their trajectory correlations, and thus significantly improves the recovery accuracy when compared to the single user recovery. Currently, compressive sensing (CS) [7] [11] is an advanced method for collective recovery in diverse applications [34] [33] [19] [36]. For trajectory recovery, CS-based social recovery [25] also produces the near-optimal approximation for the missing location data. Although CS provides high accuracy, it requires data transmission and a computing server, and thus degrades user privacy.

B. Trajectory privacy

Existing trajectory privacy works have three primary methods: *anonymization*, *dummification*, and *obfuscation*. First, a mobile user adopting anonymization method [22] is to transmit her location data attached with an anonymity instead of her ID. However, latest studies [12] [32] reveal that only anonymization is inadequate to preserve the privacy well. Second, a user adopting dummification method [18] is to transmit her location data with a set of generated fake data. Although the dummification increases the privacy, it also introduces additional data and influence the original correlations, which decreases the recovery accuracy. Third, the obfuscation method either perturbs a user's location data by mixing several other users' trajectories [15] or cloaks the data into a spatial region [13]. Existing obfuscation methods blur the original data, and thus contradict with the consideration of trajectory recover accurately.

VIII. Conclusion

With the increasing popularity of location-based service through mobile devices, it is important to simultaneously consider the quality of service and users' privacy. Focus on the trajectory recovery service in mobile social networks, in this paper, we have designed a novel privacy-preserving compressive sensing (PPCS) method to recover the user trajectories with the consideration of user privacy. The core design of PPCS is an encryption approach KVP, which leverages the matrix transformation to including privacy preservation into compressive sensing. With KVP, user trajectories are encrypted with private encrypting keys, and only encrypted data are available at the server. In this way, high level privacy preservation is provided against both eavesdroppers and stalkers. Through extensive mobility trace-based simulations, we demonstrate that PPCS not only effectively preserves the user privacy, but also achieves comparable accuracy as the state-of-the-art CS method. Although the application scenario of trajectory recovery is adopted in this work, the proposed PPCS can be utilized in other privacy-preserving data recovery applications.

References

[1] GeoLife Data Collected by Microsoft Research Asia. http://research.microsoft.com/en-us/projects/geolife/default.aspx.

[2] SUVnet Data Collected by Shanghai Jiao Tong University. http://wirelesslab.sjtu.edu.cn/download.html.

[3] Trippermap service in flickr. http://www.flickr.com/services/apps/5121/.

[4] V. S. Alagar. The distribution of the distance between random points. *Journal of Applied Probability*, pages 558–566, 1976.

[5] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. 2006.

[6] E. J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[7] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[8] C.-Y. Chow, M. F. Mokbel, and W. G. Aref. Casper*: Query processing for location services without compromising privacy. *ACM Transactions on Database Systems*, 34(4):24–48, 2009.

[9] J. W. Demmel and N. J. Higham. Improved error bounds for underdetermined system solvers. *SIAM Journal on Matrix Analysis and Applications*, 14(1):1–14, 1993.

[10] W. Dong, V. Dave, L. Qiu, and Y. Zhang. Secure friend discovery in mobile social networks. In *IEEE INFOCOM*, 2011.

[11] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

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

[13] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *ACM MobiSys*, 2003.

[14] C. Hegde, P. Indyk, and L. Schmidt. Approximation-tolerant model-based compressive sensing. In *ACM/SIAM SODA*, 2014.

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

[16] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[17] T. Jung and X.-Y. Li. Search me if you can: privacy-preserving location query service. In *IEEE INFOCOM*, 2013.

[18] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *IEEE ICPS*, 2005.

[19] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu. Data loss and reconstruction in sensor networks. In *IEEE INFOCOM*, 2013.

[20] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *ACM SIGMETRICS*, 2004.

[21] J. Liu, B. Priyantha, T. Hart, H. S. Ramos, A. A. Loureiro, and Q. Wang. Energy efficient gps sensing with cloud offloading. In *ACM SenSys*, 2012.

[22] C. Y. Ma, D. K. Yau, N. K. Yip, and N. S. Rao. Privacy vulnerability of published anonymous mobility traces. In *ACM MOBICOM*, 2010.

[23] D. Niculescu and B. Nath. Trajectory based forwarding and its applications. In *ACM MobiCom*, 2003.

[24] N. Poolsappasit and I. Ray. Towards achieving personalized privacy for location-based services. *Transactions on Data Privacy*, 2(1):77–99, 2009.

[25] S. Rallapalli, L. Qiu, Y. Zhang, and Y.-C. Chen. Exploiting temporal stability and low-rank structure for localization in mobile networks. In *ACM MOBICOM*, 2010.

[26] R. Rosales and S. Sclaroff. 3d trajectory recovery for tracking multiple objects and trajectory guided recognition of actions. In *IEEE CVPR*, 1999.

[27] G. Scaglia, A. Rosales, L. Quintero, V. Mut, and R. Agarwal. A linear-interpolation-based controller design for trajectory tracking of mobile robots. *Elsevier Control Engineering Practice*, 18(3):318–329, 2010.

[28] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, L. Girod, et al. Accurate, low-energy trajectory mapping for mobile devices. In *USENIX NSDI*, 2011.

[29] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis. Secure knn computation on encrypted databases. In *ACM SIGMOD*, 2009.

[30] T. Xu and Y. Cai. Feeling-based location privacy protection for location-based services. In *ACM CCS*, 2009.

[31] X. Yi, M. Kaosar, R. Paulet, and E. Bertino. Single-database private information retrieval from fully homomorphic encryption. *IEEE Transactions on Knowledge and Data Engineering*, 25(5):1125–1134, 2013.

[32] H. Zang and J. Bolot. Anonymization of location data does not work: A large-scale measurement study. In *ACM MOBICOM*, 2011.

[33] B. Zhang, X. Cheng, N. Zhang, Y. Cui, Y. Li, and Q. Liang. Sparse target counting and localization in sensor networks based on compressive sensing. In *IEEE INFOCOM*, 2011.

[34] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-temporal compressive sensing and internet traffic matrices. In *ACM SIGCOMM*, 2009.

[35] J. Zheng and L.-P. Chau. A motion vector recovery algorithm for digital video using lagrange interpolation. *IEEE Transactions on Broadcasting*, 49(4):383–389, 2003.

[36] Y. Zheng and M. Li. P-mti: Physical-layer missing tag identification via compressive sensing. In *IEEE INFOCOM*, 2013.

[37] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *ACM WWW*, 2009.

[38] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *IEEE ICDE*, 2008.

[39] J. Zhu, K.-H. Kim, P. Mohapatra, and P. Congdon. An adaptive privacy-preserving scheme for location tracking of a mobile user. In *IEEE SECON*, 2013.