Hindawi
Discrete Dynamics in Nature and Society
Volume 2017, Article ID 6814832, 11 pages
https://doi.org/10.1155/2017/6814832

Research Article

Prophet: A Context-Aware Location Privacy-Preserving Scheme in Location Sharing Service

Jiaxing Qu,[1] Guoyin Zhang,[1] and Zhou Fang[2]

[1]*College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China*
[2]*Heilongjiang Province Electronic Information Products Supervision Inspection Institute, Harbin 150090, China*

Correspondence should be addressed to Jiaxing Qu; qjx@nstc.gov.cn

Received 14 December 2016; Revised 26 February 2017; Accepted 16 March 2017; Published 8 May 2017

Academic Editor: Filippo Cacace

Location sharing service has become an indispensable part in mobile social networks. However, location sharing may introduce a new class of privacy threats ranging from localizing an individual to profiling and identifying him based on the places he shared. Although users may avoid releasing geocontent in sensitive locations, it does not necessarily prevent the adversary from inferring users' privacy through space-temporal correlations and historical information. In this paper, we design a Prophet framework, which provides an effective security scheme for users sharing their location information. First, we define fingerprint identification based on Markov chain and state classification to describe the users' behavior patterns. Then, we propose a novel location anonymization mechanism, which adopts a ε-indistinguishability strategy to protect user's sensitive location information published. Finally, experimental results are given to illustrate good performance and effectiveness of the proposed scheme.

1. Introduction

With the growing popularity of mobile devices (such as smartphones), millions of applications (or apps for short) with location-based services are available to users from app markets. Users release their location in order to experience personalized/customized services (such as friend-seeker and navigation service). However, location sharing may introduce a new class of privacy threats ranging from localizing an individual to profiling and identifying him based on the places he visits [1, 2] in mobile social networks.

Traditionally, k-anonymity [3] is the most widely used in privacy protection. It aims at protecting the user's identity, requiring that the attacker cannot infer the target user among other $k - 1$ different users. In the scope of the location privacy protection, spatial k-anonymity requires that it is undistinguishable among k points of interest (POI) [4]. One way to achieve this is through the use of dummy locations [5–7]. This technique needs to generate $k - 1$ dummy POIs and perform k queries to the location-based service (LBS) server, using the real and $k - 1$ dummy locations. Another way is cloaking [8, 9], which involves creating a region that includes k POIs and sending the cloaking region to the LBS server. However, such seemingly perfect k-anonymity-based methods almost need to establish some unreasonable assumptions. These methods typically assume an adversary that knows only some aspects of background knowledge and tries to prevent it from learning some other aspects. One can attack such privacy notions by changing either what the adversary already knows or what the adversary tries to learn. For example, dummy locations are feasible if and only if they look equally likely to be the real location from the view of the attacker. Any auxiliary information that allows to rule out any of those POIs, as having low probability of being the real location by some semantic properties, would immediately violate the privacy. Moreover, these existing methods mostly focus on "single shot" scenario, which fails to protect the privacy when applied to inference attack due to spatiotemporal correlations between the published geolocation contents.

In this paper, we investigate the issue of when and where the user can release his/her geolocation information. The goal

of our work is to let user enjoy location sharing service as much as possible while avoiding privacy risk. To this end, we present a context-aware location privacy-preserving scheme, called Prophet, where users' history location information is used to create statistical fingerprints of behavior patterns. We call a fingerprint as a distinctive feature allowing identification of certain behavior patterns. In this work, a fingerprint corresponds to a first-order homogeneous Markov chain, which represents a sequence of POIs appearing in a single direction flow of user's locations released. Based on this, Prophet is formalized as how to accurately and efficiently evaluate whether the users' published location information meets the user's privacy requirement. Furthermore, consider the real-life requirement that user use the location-based service. We propose a novel metric ε-undistinguishable to tradeoff between the desired level of privacy and the usefulness of the service provided by the LBS server.

We give formal security proof to the correctness and privacy guarantee of our mechanism. Furthermore, the extensive experiments demonstrate the validity and practicality of our scheme.

In summary, the paper makes the following contributions:

(1) We first present a context-aware location privacy-preserving scheme, called Prophet, and based on this, we propose a series of novel technologies for accurately and efficiently evaluating the risk of privacy.

(2) We propose a novel metric ε-undistinguishable to tradeoff between the desired level of privacy and the usefulness of the service provided by the LBS.

(3) We have implemented our scheme on our simulated testbed, and the extensive experiments demonstrate the validity and practicality of our scheme.

The remainder of the paper is organized as follows. Section 2 characterizes the system model and motivation and threat model briefly. In Section 3, we describe how to get behavior pattern fingerprints by Markov chain and state classification processes. Section 4 provides details on location anonymization mechanism, which is the key component in our scheme. Section 5 presents the experiment results confirming the effectiveness of the proposed mechanism based on the simulated testbed. Section 6 overviews related work, followed by the concluding remarks in Section 7.

2. Problem Definition

2.1. System Model. We begin by describing a high-level architecture for Prophet as illustrated in Figure 1, which involves three types of entities: user, Prophet, and LBS.

(i) *User*. In the context of LBSs, the user usually has location-based requirements (such as friend-searching, and navigation); simultaneously, he/she is reluctant to access the location-based service that may disclose his/her religious affiliations or personal lifestyle.

(ii) *Prophet*. Prophet, as an honest middleware server, provides (1) warning service, analyzing and mining users' behavior patterns from history location information released and, based on this, providing the early warning service when location sharing behavior of the user touches the red line of privacy, and (2) anonymity service, transforming the received user's location information through a technique called cloaking that hides the actual location by an anonymous space region.

(iii) *LBS*. LBS is an honest-but-curious server in our context. On the one hand, it acts in an "honest" fashion and correctly follows the designated protocol specification. However, it is "curious" to deduce and analyze location information so as to learn users' privacy.

In this framework, a user just sends his location to Prophet, while Prophet is just responsible for analyzing and anonymizing the location information sent by user without knowing the real query requirement. Similarly, when receiving the query with a certain anonymous space region, LBS provider just processes the user's query without learning the related privacy information from the anonymous space region (ASR for short).

2.2. Motivation and Threat Model. State-of-the-art methods of location privacy protection focus on anonymizing sensitive location information. These methods usually assume that the privacy requirements of users are constant and isolated. However, it is not a solid reason in the real-life location-based service scenario. For example, Bob, suffering from chronic bacterial prostatitis, is convalescing in a certain urology hospital, and he does not want anyone to know he has been to the hospital. To this end, he never checks in at this hospital. However, he may be happy to share his location by MSN to meet his friends at nearby bars or cafes where he thought no location privacy would be divulged. However, when combining Bob's check-ins and patterns of other users who have the similar behaviors, an adversary still can infer Bob's privacy. As illustrated in Figure 2, an adversary may learn that most other users follow path 3 to the hospital after leaving the bar or the cafe. During this period, even if Bob did not share any location information at hospital, the adversary can still infer that Bob probably suffers from a kind of urological disease.

3. Behavior Pattern Fingerprints

3.1. Fingerprint Identification Based on Markov Chain. In this subsection, we propose a method based on first-order homogeneous Markov chain to model possible sequences of users' behavior patterns. The benefits of using the first-order homogeneous Markov chain model are threefold: (1) it is effective enough; (2) it is simple for implementation; (3) it is easy to extend to any higher-order Markov chain model [10, 11]. We consider discrete-time random variable R_t as a first-order Markov chain for any $t = t_0, t_1, \ldots, t_n \in T$. It takes values $V_t \in \{1, \ldots, m\}$, where V_t is a decimal code of a certain POI (e.g., 9 for the store).

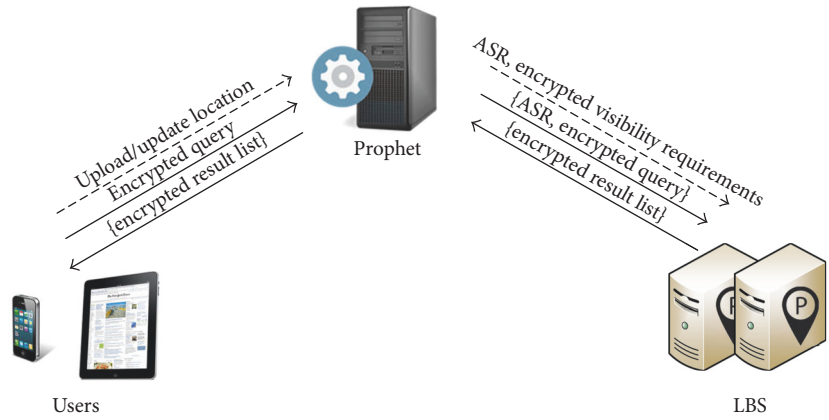

Figure 1: Architecture of allocating customer applications.

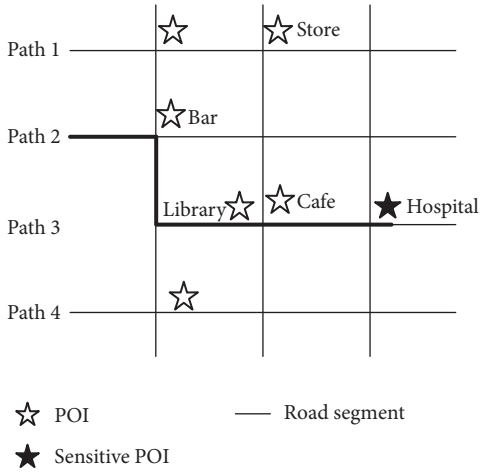

Figure 2: Architecture of allocating customer applications.

As R_t is a first-order Markov chain, we have

$$P\left(R_t = V_t \mid R_{t-1} = V_{t-1},\ R_{t-2} = V_{t-2}, \ldots, R_1 = V_1\right) = P\left(R_t = V_t \mid R_{t-1} = V_{t-1}\right). \quad (1)$$

Moreover, we further assume that the first-order Markov chain is homogeneous; that is, a state transition from time $t-1$ to time t is time-invariant, as shown below:

$$P\left(R_t = V_t \mid R_{t-1} = V_{t-1}\right) = P\left(R_t = j \mid R_{t-1} = i\right) = p_{ij}, \quad (2)$$

with the transition matrix

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m1} & P_{m2} & \cdots & p_{mm} \end{bmatrix}, \quad (3)$$

where $\sum_{j=1}^{m} p_{ij} = 1$. We denote the Input Probability Distribution (IPD) by

$$Q = [q_1, q_2, \ldots, q_m], \quad (4)$$

where $q_i = P(R_t = i)$ at time t_0, and we define

$$W = [w_1, w_2, \ldots, w_m], \quad (5)$$

as the Output Probability Distribution (OPD), where w_i denotes the probability that the location share operation (at one cycle, such as one day) finishes when it is in state i at time t_n. Note that IPD and OPD are independent in the Markov chain, which represent the probabilities to enter and leave the Markov chain. In traditional Markov chain models [11], there is an initial state and one or several ending states. In our case, IPD defines the probability to enter the state of the Markov chain, and OPD expresses the probability of aborting/leaving the Markov chain from the state set. According to these definitions, the resulting probability that a sequence of states $\{R_1, \ldots, R_T\}$ representing a behavior cycle occurs is as follows:

$$P\left(\{R_1, \ldots, R_T\}\right) = q_{V_1} \times \prod_{t=2}^{T} p_{V_{t-1}V_t} \times w_{V_T}. \quad (6)$$

The resulting probability indicates how a given sequence of location information during a state transition chain is close to one user's behavior pattern, where the larger value means that the behavior trace is closer to the model.

To illustrate the process of the fingerprint creation, consider the examples in Figures 3 and 4 of the behavior pattern sequences observed during behavior cycles in a training location information composed of only three users' behavior traces in one cycle.

There are 7 different Markov states in the example, as shown in Figure 3. The transition probability between states is derived from frequencies observed in the sequences, for example, $P_{(9:101;\ 5:104)\to(13:110)} = 92.6\%$, $P_{(10:105)\to(13:110)} = 95.2\%$. The probabilities are the parameters of the Markov chain fingerprint for the example in Figure 4. Based on this model, we can find the probability that an observed user would appear in one place based on the behavior sequences.

3.2. State Classification. In this subsection, we describe the state classification technique, which is the preliminary work of the fingerprint identifying.

Decimal code	POIs
5	Bar
9	Store
10	Library
12	Hospital
13	Movie theatre
15	Restaurant

Decimal code	Event type
101	Shopping
102	Meeting friends
103	Drinking
104	Afternoon tea
105	Reading
106	Health counseling
107	Playing game
108	Dating
109	Eating
110	Seeing film

(9: 101; 5: 104) ⟶ (13: 110) ⟶ (15: 109)
(9: 101; 5: 104) ⟶ (10: 105) ⟶ (13: 110) ⟶ (15: 109)
(5: 102) ⟶ (15: 109)
(5: 103)
(9: 101) ⟶ (5: 103)

Figure 3: An example of decimal codes for POIs and event type.

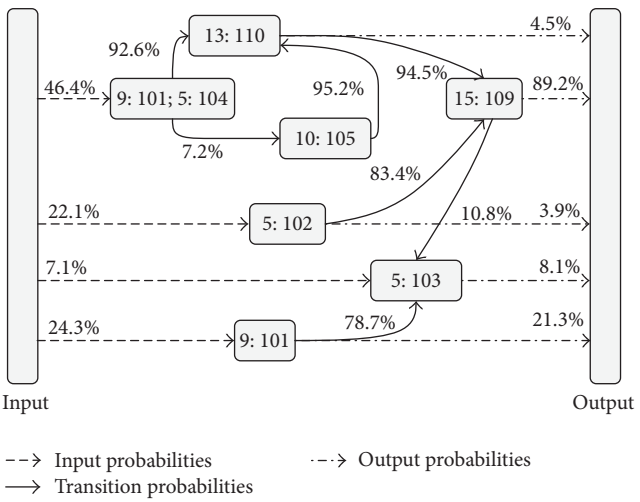

Figure 4: An example of the fingerprint for users' states.

The released location information is organized in the form of record, where each record contains the whole ordinal published geolocation contents of the corresponding user for one day. Such data is a kind of set-valued data which is sparse and high dimensional. The core idea is to find a set of states which can be used to classify into different clusters.

To this end, we introduce a data structure, named concept [12]. Given a dataset with users shared location information $D(U, A, F)$, where $U = \{x_1, \ldots, x_n\}$, each x_i $(i \leq n)$ is a shared information which records a user shared location-content sequence for one day; $A = \{a_1, \ldots, a_m\}$, each a_j $(j \leq m)$ is a state; $F : U \times A \to V \in [0, 1]$, the concept set is denoted by $C(I, E, h, g)$, where I denotes the state set, called the intension of the concept; E is the corresponding records, called the extension of the concept; and h and g are a pair of dual operators, defined by, for $I \subseteq A$ and $E \subseteq U$,

$$h(E) = \{a \in A \mid (x, a) \in F \ \forall x \in U\},$$
$$g(I) = \{x \in U \mid (x, a) \in F \ \forall a \in A\}. \quad (7)$$

Specifically, when I contains sensitive states A_S $(A_S \neq \emptyset, A_S \subseteq A)$, we called the corresponding concept as privacy concept C_S; otherwise, the concept is called information concept C_I.

Based on this, we can see that the problem of identifying the core states can be reduced into the mapping relation from information concept C_I to C_S. Specifically, we first partition dataset into several parts horizontally, according to the value difference of I_S. Then, focusing on the extension set of $C_S(I_S, E_S)$ by the operation $g(I_S)$, we need to find all information concepts $\mathbb{C}_I$ with the intension set I_S. Here, to prevent the dimension disaster caused by the sparse and high dimensional state set, we use two parameters (k, m) as the threshold of state aggregation, where k denotes the minimum support threshold of the states and m denotes the confidence threshold of the state aggregation, which requires that, focusing on a state aggregation, any m state as a whole meets the minimum support threshold.

Lemma 1 (a priori property). *Given a concept $C(I, E)$ over $D(U, A, F)$, where $I \neq \emptyset$ and $|E| \geq k$, for any concept $C_i(I_i, E_i)$ with $I_i \subseteq I$ $(I_i \neq \emptyset)$, we have $|E_i| \geq k$.*

There is a fact that if a concept $C_i(I_i, E_i)$ does not meet the support threshold k, all of the higher-dimensional concepts $C(I, E)$ with $(I \supset I_i)$ will not meet the support threshold as well. According to Lemma 1, generating the i-dimensional concept set

$\mathbb{C}_i$ ($i > 1$) just needs the previous concept set $\mathbb{C}_{i-1}$. Specifically, given the threshold parameters (k, m), we generate all of i-dimensional concept set ($i \le m$) iteratively as follows:

(1) *Generate candidate 1-dimensional concept set. Each state constitutes the intension of a candidate 1-dimensional concept. The algorithm scans all of the records in the target cluster, recording the corresponding extension and the size of the extension domain.*

(2) *Generate $\mathbb{C}_1$. Based on the threshold k, the 1-dimensional concept set $\mathbb{C}_1$ can be determined. It consists of the candidate 1-dimensional concept set, where $|E_j|$ of each concept C_j is equal to or greater than k.*

(3) *Generate i-dimensional concept set $\mathbb{C}_i$ ($i > 1$) based on the concept set $\mathbb{C}_{i-1}$. The algorithm first uses the join $\mathbb{I}_{i-1} \bowtie \mathbb{I}_{i-1}$ to generate a candidate i-dimensional intension set. Then, based on the a priori property (Lemma 1) that all subconcepts of a higher-dimensional concept satisfying the threshold also satisfy the support threshold, we can prune the candidate intension sets that do not satisfy the a priori property. For each of the rest candidate intension sets, we compute the intersection of extension sets corresponding to $(i-1)$-dimensional concepts. Then, the i-dimension concept set can be determined. It consists of the concepts, where $|q(\mathbb{I}_i)|$ of $\mathbb{C}_i$ is equal to or greater than k.*

(4) *Jump to Step (3) until $\mathbb{C}_i = \emptyset$ or $i = m$.*

We can see that, given the parameters (k, m), the issue of finding all information concepts $\mathbb{C}_I$ with the intension set I_S is transferred to the m rounds concept-generating.

Next, we say that one privacy concept C_S is a domain. By building the discernibility matrix [13] from the information concept set $\{\mathbb{C}_1, \ldots, \mathbb{C}_i\}$ to C_S, we can find the core state set.

Definition 2 (discernibility matrix). Given a domain $C_S(I_S, E_S)$ and the corresponding information concept set $\{\mathbb{C}_1, \ldots, \mathbb{C}_i\}$, the discernibility set

$$\mathrm{DS}\left(C_{Ii}, C_{Ij}\right) = \begin{cases} \left\{a_k \in A \mid E_{Ii} \neq E_{Ij}\right\} & \left(I_{Si} \cap I_{Sj} = \emptyset\right) \\ \emptyset & \left(I_{Si} \cap I_{Sj} \neq \emptyset\right). \end{cases} \tag{8}$$

We say that $\mathrm{DM} = (\mathrm{DS}(C_{Ii}, C_{Ij}) \mid I_{Ii}, I_{Ij} \in A_I)$ is the discernibility matrix.

Based on Definition 2, we can find the core state set A_c as shown in

$$A_c = \bigcap_{\mathrm{DS}(C_{Ii}, C_{Ij}) \in \mathrm{DM}} \mathrm{DS}\left(C_{Ii}, C_{Ij}\right). \tag{9}$$

4. Location Anonymization Mechanism

To protect user's location privacy from LBS provider, the Prophet would generate an anonymous space region that contains several POIs located next to the user's exact location. Normally, in the perspective of the information publisher, the bigger the ASR is, the higher the accuracy loss of the released information is. Unfortunately, this rule is not always true: an adversary can fast narrow it down by eliminating fallacious POIs. Here, we propose a novel ε-undistinguishable anonymity mechanism to solve this issue, which contains two stages: preprocessing and region-anonymizing.

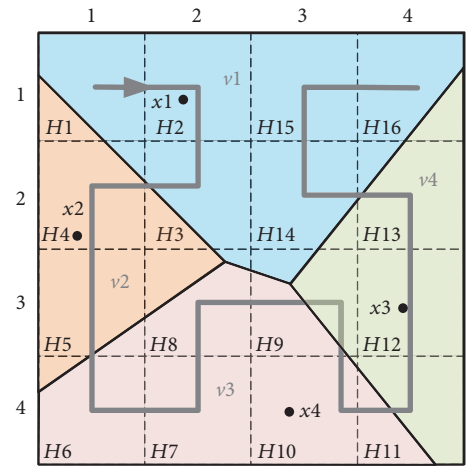

Figure 5: Computing the Voronoi diagram for the POIs.

4.1. Preprocessing. To quantitatively customize the ASR, in this stage, we need to location POIs. de Berg et al. [14] adopt the Voronoi diagram to divide the space into a set of Voronoi cells (Vcells), where each POI is assigned to a Vcell. However, it gives rise to the following problems: (a) due to the irregular ASR generated by Voronoi rule, it is difficult to be transformed by the coordinate representation. Moreover, the size of each subregion is inhomogeneous, which makes it difficult to quantitatively assess the mapping relation on the basis of generating the ASR between users' distribution and the distribution of the POIs; (b) Because the number of subregions partitioned by the Voronoi rule is equal to the number of POIs, the target region could not be subdivided. To this end, on top of the Voronoi diagram, we use Hilbert space-filling curve, which superimposes a regular $N \times N$ grid where each grid cell (Gcell) stores information about the Vcells intersecting it. The information recorded in each Gcell G_{ij} can be viewed as a tuple (X, H), where X is POIs contained in this Gcell and H is an index set that records such Gcells: (a) when $X = \emptyset$, H records the Hilbert number of such Gcells that contain POIs in the Vcells intersecting of the target Gcell G_{ij}; (b) $X \neq \emptyset$, H records all Hilbert numbers in the Vcells covering X. For example, Figure 5 shows a 4×4 grid, where G_{12} containing x_2 stores $(\{x_2\}, \{H_1, H_3, H_4, H_5, H_6, H_8\})$ and G_{13} intersected by $\{V_2, V_3\}$ stores $(\emptyset, \{H_4, H_{10}\})$, and so on.

4.2. Region-Anonymizing. k-anonymity is one of the most popular security metrics, which makes each published record undistinguishable from at least $k-1$ other records. However, in the context of LBS, k-anonymity is not so nice as it seems. Kalnis et al. [8] show a set of attacks against space k-anonymity. In this subsection, we define a novel privacy

metrics ε-indistinguishability, which expresses a user's privacy requirement and information availability, simultaneously.

In the view of privacy protection, the covered POIs (containing the selected $k-1$ POIs and the target POI) in ASR should be undistinguishable in the probability. Here, we assume that the adversary has held some auxiliary information S (the target user's previous tracks). Consider two POIs x and x'; we say x and x' are ε-undistinguishable *iff*

$$\frac{P(x)}{P(x')} = \frac{P(x \mid S)}{P(x' \mid S)} = e^{\varepsilon} \quad (\varepsilon > 0). \tag{10}$$

Intuitively, since two locations x and x' produce a reported value in S with similar transition probabilities, S reveals little information about whether the actual location is x or x'.

In the view of information availability, it is obvious that the information availability of the released location relevant content is distance-dependent. That is, given an information loss level l, it is proportional to the radius r of the ASR, more formally:

$$l(r) = \varepsilon' r \quad \left(r \gg \varepsilon' > 0\right), \tag{11}$$

where the parameter ε' can be thought as the level of information loss at one unit of distance. This definition requires that the user is protected within any radius r, but with a level l that increases with the distance.

Combining (10) and (11), we get the final definition of ε-undistinguishable.

Definition 3 (ε-indistinguishability). Assuming that the adversary has held some auxiliary information S (the target user's previous tracks), we say a mechanism satisfies ε-undistinguishable *iff* for any two POIs x and x'

$$\frac{P(x \mid S)}{P(x' \mid S)} = e^{\varepsilon/r} \quad \left(r \gg \varepsilon > 0,\ d\left(x, x'\right) \le r\right). \tag{12}$$

Based on Definition 3, we can see that (1) when $\varepsilon \to 0$, the strength of privacy protection is also strengthened gradually; (2) when r reduces gradually, information loss is also reduced but the strength of privacy protection is affected; and (3) by adjusting the parameters ε and r, the issue of building ASR which can make a tradeoff between privacy and information availability can be transformed into the following optimization problem:

$$\begin{aligned} \text{Maximize} \quad & \frac{\max_{x_i \in \text{ASR}} P(x_i \mid S)}{\min_{x_j \in \text{ASR}} P(x_j \mid S)} \\ & + \lambda \frac{1}{\left(O_{x\max} - O_{x\min}\right) \times \left(O_{y_{\max}} - O_{y_{\min}}\right)} \\ \text{subject to} \quad & \sum_{x_i \in \text{ASR}} 1 \ge k, \\ & \frac{\max_{x_i \in \text{ASR}} P(x_i \mid S)}{\min_{x_j \in \text{ASR}} P(x_j \mid S)} \le e^{\varepsilon}, \end{aligned} \tag{13}$$

```
INPUT:  k, ε // anonymity parameters
        r // information availability parameters
        q_loc // user's exact location
        T // the Hilbert threaded regions
OUTPUT: ASR // anonymous space region
BuildASR (m, k, e, q_loc, T)
(1)  q_x = q_loc;
// Step (1)
(2)  Loc = q_x;
(3)  num = Hilbert (q_loc);
(4)  do
(5)      num++;
(6)      Loc = Loc ∪ Hilbert^{-1}(num);
(7)  while |Loc| ≥ k;
(8)  for i = 1; i++; i ≤ |Loc| do
(9)      for j = i + 1; j++; i ≤ |Loc| do
(10)         E[i][j] = P(q_i | S) / P(q_j | S);
(11) for each E[i][j] in E do // Step (2)
(12)    if E[i][j] > e^ε then
(13)        record the correspond POIs q_i and q_j into Q;
(14) delete POIs of Q from Loc;
(15) num = Hilbert (median(Loc));
(16) jump to Step (1);
     // Step (3)
(17) find the points O_top-left, O_bottom-right from E;
(18) compute the area of ASR (O_top-left, O_bottom-right);
(19) if ASR > 4r^2 then
(20)    delete POIs corresponding with O_top-left, O_bottom-right;
(21) num = Hilbert (median(Loc));
(22) jump to Step (1);
(23) return ASR;
```

Algorithm 1: Building the ASR.

$$\left(O_{x\max} - O_{x\min}\right) \times \left(O_{y_{\max}} - O_{y_{\min}}\right) \le 4r^2, \tag{14}$$

where $(O_{x\max} - O_{x\min}) \times (O_{y_{\max}} - O_{y_{\min}})$ are the area of the ASR and λ is a nonnegative weight.

Obviously, this is a NP-complete problem which can be reduced into the 0-1 knapsack. Therefore, we propose a heuristic algorithm, as follows.

Algorithm 4. Before describing the details of the proposed algorithm, we first introduce its core idea. When receiving the location sharing requirement, Prophet first checks user's behavior fingerprint. If the shared location belongs to "the core state set" and the probability of inferring sensitive/privacy location based on the computation of the transition matrix is greater than the preset threshold, Prophet would issue an alarm to the user. After getting the response from the user, Prophet builds the corresponding ASR satisfying the privacy requirement and information availability. The heuristic rule of building the ASR is shown in Algorithm 1: Step (1): locating the Hilbert number of the shared location, the algorithm traverses space regions alone of the Hilbert space-filling curve until finding $k-1$ neighboring POIs. And then it computes the corresponding privacy strength and the area of the ASR. If user's requirements are satisfied,

the algorithm terminates; otherwise, three conditions are discussed. Step (2) (Condition 1: privacy strength is lower than the threshold): eliminating the POIs (x_i and x_j) whose $P(x_i \mid S)/P(x_j \mid S) > e^{\varepsilon}$, it finds the median of the Hilbert number among the rest of POIs, and based on the median, it adds two different neighboring POIs which is similar to Step (1). Step (3) (Condition 2: the area of ASR is greater than the threshold): eliminating two POIs (the one having the biggest abscissa value and the one having the biggest ordinate value), it finds the median of the Hilbert number among the rest POIs, and based on the median, it adds two different neighboring POIs which is similar to Step (1). Step (4): it iteratively performs Steps (2) and (3) until satisfying user's requirement or aborting due to being unable to find a convergence of the solution space.

4.3. Security Analysis. Due to introducing Prophet as a trusted third party (TTP for short), there is no collusion attack from Prophet and LBS server. Based on right decentralization mechanism, LBS server cannot accurately infer the sensitive location hided by user. Furthermore, against inference attack, Prophet adopts two-stage privacy protection strategy: Markov chain-based reverse inference mechanism (Section 3) and location anonymization mechanism (Section 4). The strategy proposed in Section 3 can estimate the probability that the adversary infers user based on the published check-in chains. Based on this, the strategy proposed in Section 4 further anonymizes user's location before publishing. Based on the proposed region-anonymizing mechanism, the adversary cannot infer more privacy information than the published one. Integrating the proposed two anonymous strategies, we are able to assess the probability that the adversary infers the sensitive location hidden by user. Assume that adversary has known the check-in chain containing m regions (k POIs for each region); the final inference probability P is as follows:

$$P = \prod_{i=1}^{m-1} p_{i \to i+1} \times k_{i \to i+1} \times \alpha_{i \to i+1}, \tag{15}$$

where $p_{i \to i+1}$ denotes the inference probability that the target user checks in from POI i to POI $i + 1$; $k_{i \to i+1}$ denotes the number of POIs contained in the ith region; $\alpha_{i \to i+1}$ is the attenuation ration; and α decreases along with the check-in chain.

5. Evaluation and Experiment

We now evaluate some performance results of our scheme using real-world dataset, Foursquare, made available by Gao et al. [15]. It contains the check-in history of 18107 users ranging from March 2010 to January 2011. Our simulated testbed is implemented on a workstation with 2 Intel Xeon E3 core processors running at Intel 2.13 GHz CPUs, 32 GB dual-channel 1333 GHz memory for Prophet server and LBS provider server, respectively. We report the performance and effectiveness of the proposed anonymity algorithm, respectively. The implementation for the proposed algorithms uses Python.

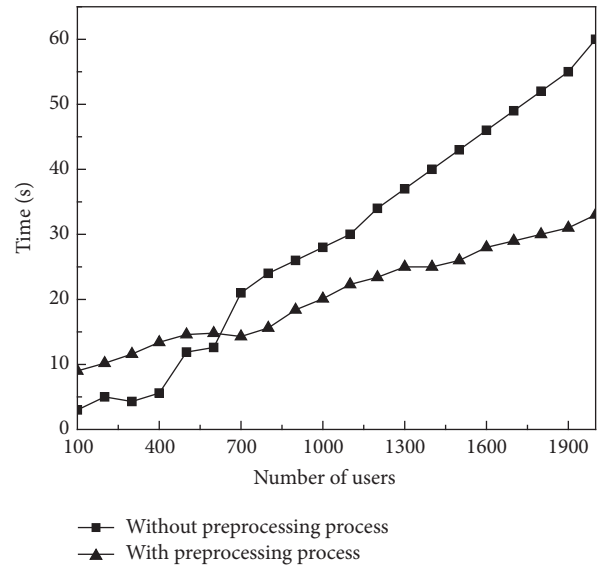

Figure 6: The overhead of building transition matrix.

5.1. Building Transition Matrix. As mentioned above, transition matrix is the core preliminary work of proposed location anonymization scheme. Hence the overhead in building transition matrix phase directly affects the whole scheme. Now, we begin by estimating the cost in terms of building transition matrix. Suppose that the number of users varies from 100 to 2,000, in steps of 100, in the following experiment. Under this setting, we quantify the cost introduced by the building transition matrix in terms of fingerprint identification as well as state classification, as shown in Figure 6.

The experimental results in Figure 6 show the overhead in building transition matrix with varying numbers of users. For comparison, we include a direct scheme of building transition matrix as a baseline, which does not contain the step of state classification. We also can see that the overhead of building transition matrix in state classification phase increases, as the number of users increases compared with the direct scheme.

Specifically, there are only 60.12 seconds in building transition matrix phase for 2000 users. This experimental results demonstrate the effectiveness of proposed state classification phase by concept data structure. In other words, this overhead is acceptable, even for very large number of users. This result demonstrates the basic usability of our scheme for fingerprint identification calculating phase.

5.2. Building ASR. As discussed in Section 4, the overhead of building ASR is closely related to parameters r and ε. Hence, we evaluate this effectiveness through multigroup experiments. Then the next group of experiments illustrate the performance of the proposed anonymity scheme from the following phases, where k is 3, 5, 7, 9, and 11.

Figure 7 shows the execution timings of building ASR as $r = 1$ km. Obviously, the overhead of building ASR grows slowly on different ε value. With the increasing of ε, the overhead of building ASR increases gradually. This result confirms the effectiveness of our behavior pattern fingerprints

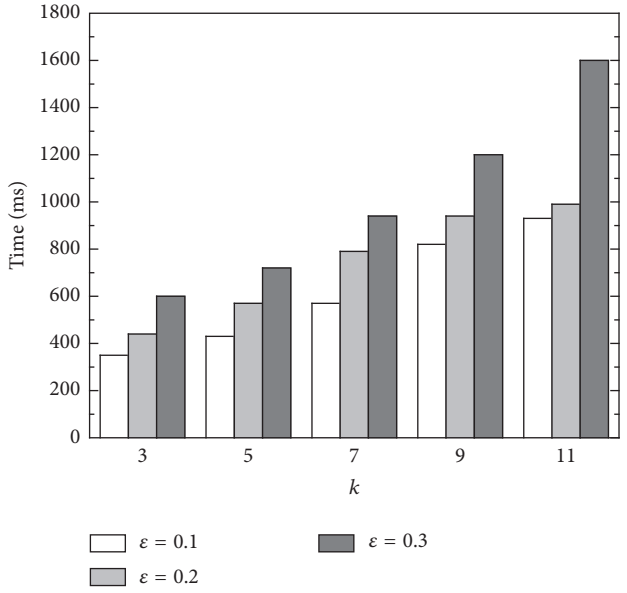

Figure 7: The overhead of building ASR on different ε.

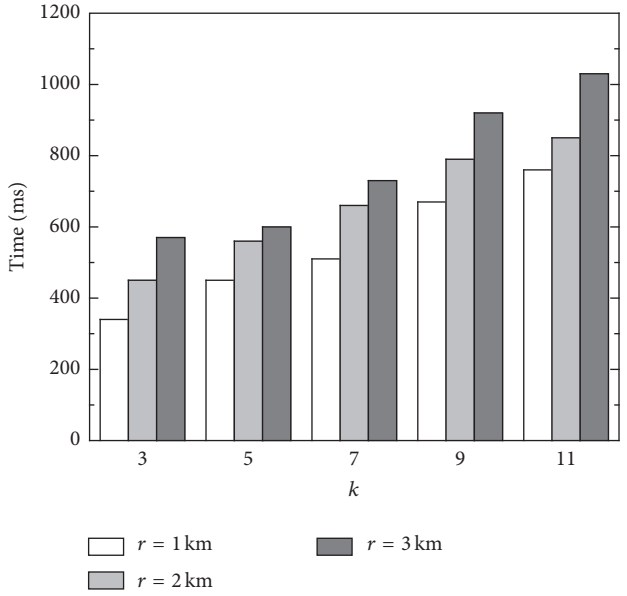

Figure 8: The overhead of building ASR on different r.

recognition scheme, which is the core preliminary work for the proposed region-anonymizing scheme.

On the other side, Figure 8 shows the overhead of building ASR experimental results where $\varepsilon = 0.1$. Obviously, it takes much more time to build ASR with r increasing. For example, it only takes 1030.4 *ms* to construct ASR by Prophet, where $k = 11$ km and $r = 3$ km. The main reason of low computation overhead in ASR building phase is that the preprocessing process normalizes the pattern fingerprints for each user and performs excellently for classification.

In short, the overhead of building ASR does not introduce much more negative impact on the whole scheme by different r and ε. That is because the preprocessing phase mainly

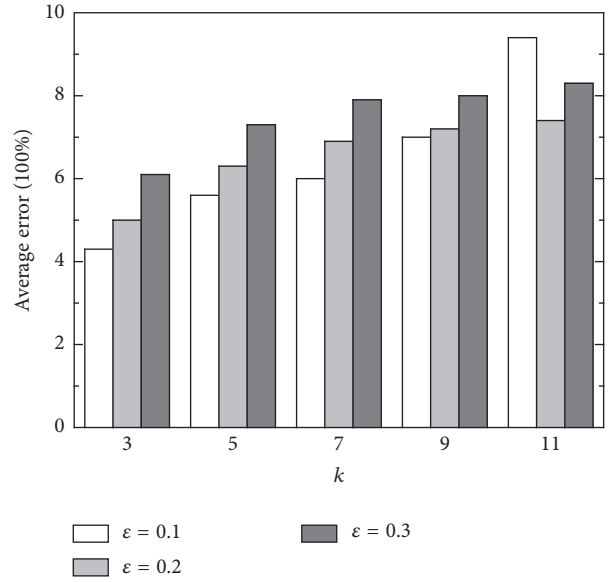

Figure 9: The average error of Prophet on different ε.

focuses on minimizing the computation overhead in building ASR process. Hence, the proposed location anonymization mechanism releases Prophet from heavy computational overhead in building ASR phase, which satisfies real-world situations.

5.3. Effectiveness. Next, we focus on evaluating the performance of our privacy-preserving scheme during the preprocessing and anonymizing procedure. As discussed in Section 4, the proposed location anonymization scheme is a heuristic algorithm. That is, the constructed ASR may not be the optimal one in theory. Therefore, we calculate the average error of our scheme compared to theoretical value through 100 simulated experiments.

Figure 9 plots the average error of our proposed scheme on different ε values. As it can be seen, the higher average error is only 8.33% where $k = 11$ and $\varepsilon = 0.3$. One important reason of this result is that our proposed region-anonymizing heuristic algorithm can find the optimal value effectively. Compared to the theoretical value, the high accuracy of the proposed scheme can be proved directly. Figure 10 shows the average error of our proposed scheme on different r values; similar to the above experiment, the average error is not very high (about 7%), which also satisfies the real-world situations.

5.4. Performance. As previously mentioned, the real sensitive locations are usually hidden by users in our datasets Foursquare. In order to evaluate the location indistinguishability among Prophet, CLPP [16], and DP [17], we selected two sensitive locations in our next experiment as illustrated in [16]. Two hidden locations sets, HL_1 and HL_2, are generated by randomly marking off a portion of POIs and adding POIs which are geographically located between the POIs, respectively.

There are two metrics designed to evaluate the accuracy among Prophet, DP, and CLPP: (1) average confidence of

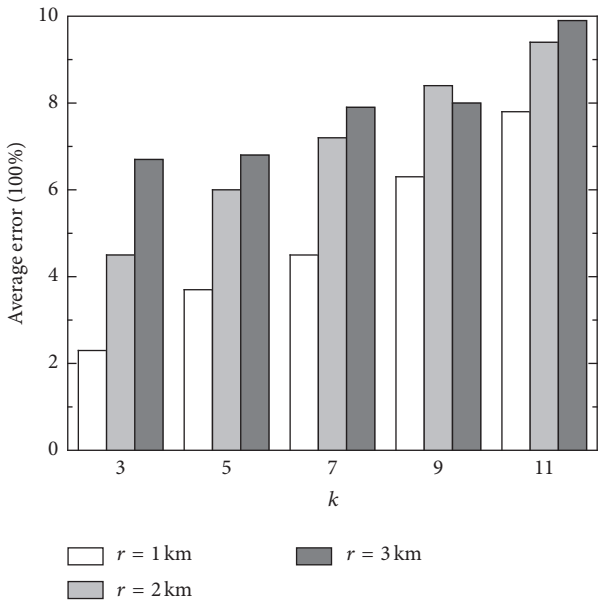

Figure 10: The average error of Prophet on different r.

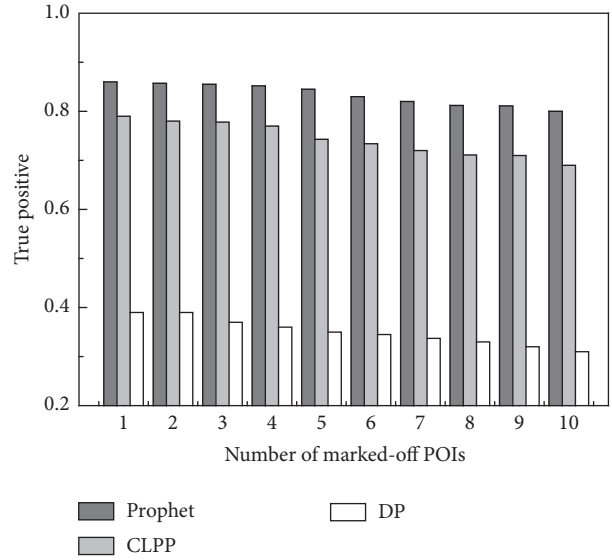

Figure 11: Performance evaluation on true positive probability.

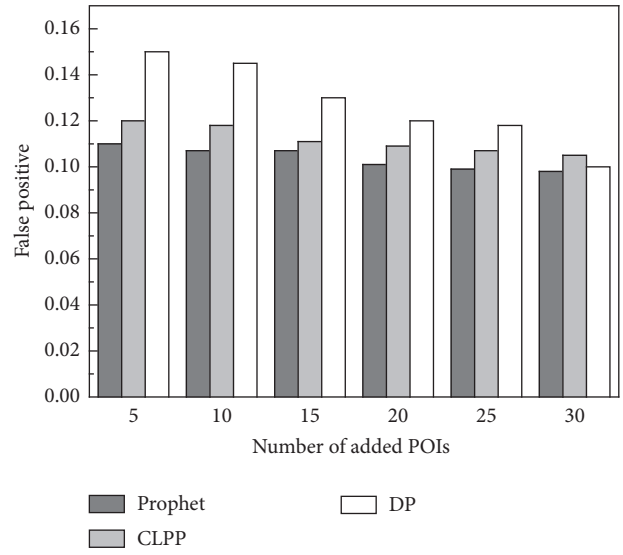

Figure 12: Performance evaluation on false positive probability.

hidden location set HL_1, denoted as true positive, and (2) average confidence of hidden location set HL_2, denoted as false positive [16]. In detail, we select 1,000 users and choose l_1 and l_2 in each user's check-in history records. We randomly mark off $1, 2, 3, \ldots, 10$ POIs between l_1 and l_2 as HL_1 and add 5, 10, 15, 20, 25, and 30 un-checked-in POIs between l_1 and l_2 as HL_2.

The experimental results shown in Figures 11 and 12 demonstrate that Prophet has better performance than CLPP and DP in terms of true positive probability and false positive probability under all experimental values. This is because some users' check-in historical data are always personal and unusual, which makes it difficult for CLPP and DP to evaluate whether the user has visited the hidden locations within large amounts of users' historical check-in historical data in Foursquare. It is important to note that the higher the true positive probability in Figure 11 is, the better the scheme is, while false positive probability in Figure 12 shows the opposite. Those results further denote that our proposed region-anonymizing strategy performs quite well as discussed in Section 4.3. Meanwhile, we can conclude that the increasing number of marked-off or added POIs does not seriously affect the true confidence or false confidence of Prophet.

6. Related Work

Privacy-preserving has attached much more attentions in mobile social networks research field areas. Most of current privacy-preserving schemes which focus on sensitive data sharing issues are dependent on anonymization techniques [16–19] or cryptographic algorithm [20]. CLPP is used to evaluate whether the users' published location information meets their privacy requirement in location-based social network through traditional mining algorithm [16]. However, CLPP is not sufficient to ensure user's location privacy due the weak classifier of mining algorithm compared to our proposed Prophet. To address data sparsity problem, DP strategy selects a minimum number of locations a user has to hide on the trajectory by subtrajectory synthesis algorithm in order to avoid privacy leakage risk [17]. But unlike CLPP and DP, in our work, $\boldsymbol{\varepsilon}$-indistinguishability strategy based on fingerprint identification is a novel aspect of this work. Bilogrevic et al. [19] propose a privacy-preserving method for mobile devices to the server based scheduling problem which takes full use of the homomorphic properties of asymmetric cryptosystems to calculate common user availabilities in order to meet user's personalized privacy requirements. Different from traditional privacy-preserving research in cloud environment, several researches focus on methodologies for the implementation of context-aware environment in mobile

cloud. Lin et al. [18] provided a reliable recommendation and privacy-preserving based cross-layer reputation mechanism (RP-CRM) to provide secure and privacy-aware communication process in mobile cloud environment. Chen et al. [21] discussed how to use local trust value, which is calculated based on user call behavioral attributes in order to protect user's sensitive behavior patterns of mobile cloud user. Those works focus on privacy-preserving research on DaaS (Data as a Service) and privacy-aware communication in cloud. Biswas and Vidyasankar [20] resort to integrating transactional and cryptographic primitive scheme to realize privacy-preserving of sensitive data against untrusted cloud servers. Reference [22] used a range of applications of Virtual Individual Servers (VIS) proxies to protect mobile device privacy. However, different from traditional method, [23] provides oblivious transfer and private information retrieval interaction scheme to achieve an efficient and practical location-based privacy-preserving problems based on queries.

Some research works focus on the privacy-preserving of healthcare information in mobile health monitoring environment [24, 25]. Cloud-assisted mobile health (mHealth) monitoring is a revolutionary way to improve the quality of healthcare service. However, this situation poses a serious risk on both clients' privacy and intellectual property. Cloud-assisted mHealth monitoring system (CAM) [24] which relies on the anonymous Boneh-Franklin identity-based encryption (IBE) in medical diagnostic programs. SPOC [25] is a secure and privacy-preserving opportunistic computing framework which is based on attribute-based access control and a new privacy-preserving scalar product computation (PPSPC) technique to protect the users' personal health information (PHI) security.

The study of location-based anonymize scheme has gained the great interest from the research community recently, and we briefly review some of them related to our work [26–31]. In [26], users' location is encrypted when shared in mobile social applications and can be only decrypted by the data owner. In [27], the credential information is updated on the basis of mobile cloud packets exchange, protecting against credential faking or stealing attacks. MobiShare [28] is a location privacy framework in mobile online social networks by separating user identities and anonymized location updates. Secure mobile user-based data service mechanism (SDSM) [29] provided confidentiality access control for data stored in the cloud by identity-based proxy reencryption scheme. FINE [30] employed a ciphertext-policy anonymous attribute-based encryption technique to achieve location privacy for mobile devices. FindU [31] which is a set of privacy-preserving distributed profile matching schemes in mobile social networks resorts to Shamir secret sharing as the main secure computing technique. Although these schemes solve location anonymizing problem in mobile cloud, they do not emphasize how to transfer the workload of the involved parties to the cloud without violating the privacy of involved parties. Since our scheme scenario contains preprocessing phase, it is helpful to release heavy computational load on Prophet in behavior pattern fingerprints phase.

7. Conclusion

In this paper, we design a context-aware location privacy-preserving scheme in mobile cloud environment, named Prophet, which is an effective security scheme for mobile cloud users to protect the mobile user's sharing locations. Moreover, we propose a novel location anonymization mechanism, which adopts a ε-indistinguishability strategy to protect user's sensitive location information published. In addition, through extensive performance evaluation, we have also demonstrated that Prophet can balance the privacy requirement and acceptable information availability.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] A. N. Khan, M. L. Mat Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: a survey," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1278–1299, 2012.

[2] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: a survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.

[3] L. Sweeney, "*k*-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[4] I.-T. Lien, Y.-H. Lin, J.-R. Shieh, and J.-L. Wu, "A novel privacy preserving location-based service protocol with secret circular shift for k-NN search," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 863–873, 2013.

[5] B. Niu, Q. Li, X. Zhu et al., "Achieving k-anonymity in privacy-aware location-based services," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '14)*, Toronto, Canada, April-May 2014.

[6] W. Yao, P. Ye, and X. Li, "An effective privacy-preserving algorithm based on logistic map and rubik's cube transformation," *Discrete Dynamics in Nature and Society*, vol. 2014, Article ID 178585, 2014.

[7] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing privacy through caching in location-based services," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications (IEEE INFOCOM '15)*, pp. 1017–1025, IEEE, May 2015.

[8] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1719–1733, 2007.

[9] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference Mobile Systems, Applications, and Services (MobiSys '03)*, pp. 31–42, San Francisco, Calif, USA, May 2003.

[10] S. R. Eddy, "Hidden Markov models," *Current Opinion in Structural Biology*, vol. 6, no. 3, pp. 361–365, 1996.

[11] M. Korczyński and A. Duda, "Markov chain fingerprinting to classify encrypted traffic," in *Proceedings of the 33rd IEEE Conference on Computer Communications (IEEE INFOCOM '14)*, pp. 781–789, May 2014.

[12] H. Zhang, Z. Zhou, L. Ye, and X. Du, "Towards privacy preserving publishing of set-valued data on hybrid cloud," *IEEE Transactions on Cloud Computing*, 2015.

[13] Z. Zhou, H. Zhang, X. Du, P. Li, and X. Yu, "Prometheus: privacy-aware data retrieval on hybrid cloud," in *Proceedings of the 32nd IEEE Conference on Computer Communications (IEEE INFOCOM '13)*, pp. 2643–2651, April 2013.

[14] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, Springer, 2nd edition, 2000.

[15] H. Gao, J. Tang, and H. Liu, "Exploring social-historical ties on location-based social networks," in *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (ICWSM '12)*, pp. 114–121, Dublin, Ireland, June 2012.

[16] H. Zhang, Z. Xu, Z. Zhou, J. Shi, and X. Du, "CLPP: context-aware location privacy protection for location-based social network," in *Proceedings of the IEEE International Conference on Communications (ICC '15)*, pp. 1164–1169, IEEE, London, UK, June 2015.

[17] A. Y. Xue, R. Zhang, Y. Zheng, X. Xie, J. Huang, and Z. Xu, "Destination prediction by sub-trajectory synthesis and privacy protection against such prediction," in *Proceedings of the 29th International Conference on Data Engineering (ICDE '13)*, pp. 254–265, April 2013.

[18] H. Lin, L. Xu, Y. Mu, and W. Wu, "A reliable recommendation and privacy-preserving based cross-layer reputation mechanism for mobile cloud computing," *Future Generation Computer Systems*, vol. 52, article no. 2655, pp. 125–136, 2015.

[19] I. Bilogrevic, M. Jadliwala, P. Kumar et al., "Meetings through the cloud: privacy-preserving scheduling on mobile devices," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1910–1927, 2011.

[20] D. Biswas and K. Vidyasankar, "Privacy preserving and transactional advertising for mobile services," *Computing*, vol. 96, no. 7, pp. 613–630, 2014.

[21] S. Chen, G. Wang, and W. Jia, "A trust model using implicit call behavioral graph for mobile cloud computing," in *Cyberspace Safety and Security*, pp. 387–402, Springer, 2013.

[22] R. Cáceres, L. Cox, H. Lim, A. Shakimov, and A. Varshavsky, "Virtual individual servers as privacy-preserving proxies for mobile devices," in *Proceedings of the 1st ACM Workshop on Networking, Systems, and Applications for Mobile Handhelds*, pp. 37–42, IEEE, Barcelona, Spain, August 2009.

[23] R. Paulet, M. G. Kaosar, X. Yi, and E. Bertino, "Privacy-preserving and content-protecting location based queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1200–1210, 2014.

[24] H. Lin, J. Shao, C. Zhang, and Y. Fang, "CAM: cloud-assisted privacy preserving mobile health monitoring," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 985–997, 2013.

[25] R. Lu, X. Lin, and X. Shen, "SPOC: a secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 3, pp. 614–624, 2013.

[26] K. P. N. Puttaswamy and B. Y. Zhao, "Preserving privacy in location-based mobile social applications," in *Proceedings of the 11th Workshop on Mobile Computing Systems and Applications (HotMobile '10)*, pp. 1–6, February 2010.

[27] A. N. Khan, M. L. M. Kiah, S. A. Madani, A. U. R. Khan, and M. Ali, "Enhanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing," *The Journal of Supercomputing*, vol. 66, no. 3, pp. 1687–1706, 2013.

[28] W. Wei, F. Xu, and Q. Li, "MobiShare: flexible privacy-preserving location sharing in mobile online social networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 2616–2620, March 2012.

[29] W. Jia, H. Zhu, Z. Cao, L. Wei, and X. Lin, "SDSM: a secure data service mechanism in mobile cloud computing," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS '11)*, pp. 1060–1065, IEEE, Shanghai, China, April 2011.

[30] J. Shao, R. Lu, and X. Lin, "FINE: a fine-grained privacy-preserving location-based service framework for mobile devices," in *Proceedings of the IEEE INFOCOM*, pp. 244–252, IEEE, Ontario, Canada, May 2014.

[31] M. Li, S. Yu, N. Cao, and W. Lou, "Privacy-preserving distributed profile matching in proximity-based mobile social networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2024–2033, 2013.