A Caching and Spatial K-anonymity Driven Privacy Enhancement Scheme in Continuous Location-Based Services

Shaobo Zhang[a,b], Xiong Li[a], Zhiyuan Tan[c], Tao Peng[d], Guojun Wang[d,*]

[a]*School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, 411201, China*
[b]*Collage of Computer, National University of Defense Technology, Changsha, 410073, China*
[c]*School of Computing, Edinburgh Napier University, Edinburgh, EH10 5DT, United Kingdom*
[d]*School of Computer Science and Technology, Guangzhou University, Guangzhou, 510006, China*

Abstract

With the rapid pervasion of location-based services (LBSs), protection of location privacy has become a significant concern. In most continuous LBSs' privacy-preserving solutions, users need to transmit the location query data to an untrusted location service provider (LSP) to obtain query results, and the users discard these results immediately after using them. This results in an ineffective use of these results by future queries and in turn leads to a higher risk to user privacy from the LSP. To address these issues, we generally use caching to cache the query results for users' future queries. However, the minimization of the interaction between users and LSPs is a challenge. In this paper, we propose an enhanced user privacy scheme through caching and spatial K-anonymity (CSKA) in continuous LBSs; it adopts multi-level caching to reduce the risk of exposure of users' information to untrusted LSPs. In continuous LBS queries, our scheme first utilizes the Markov model to predict the next query location according to the user mobility. Then, according to the predicted location, cell's cache contribution rate, and data freshness, an algorithm for forming spatial K-anonymity is designed to improve the user's cache hit rate and enhance the user location privacy. The security analysis and simulation results demonstrate that our proposed CSKA scheme can provide higher privacy protection than a few previous methods, and it can minimize the overhead of the LBS server.

Keywords: Location privacy, Multi-level caching, Spatial K-anonymity, User mobility, Cache hit rate

1. Introduction

With the proliferation of wireless communication and mobile phones featuring global positioning system (GPS), location-based services (LBSs) have attracted substantial interest and are becoming one of the fastest-growing services [1, 2]. In an LBS, a mobile user sends a request containing his/her location and interests to a location service provider (LSP). The LSP returns the point of interests (POIs) near the user's current location (e.g., garages, car parks, supermarkets, and restaurants) according to the user's interests. However, the LSP has

*Corresponding author
Email address: csgjwang@gmail.com (Guojun Wang)
URL: http://trust.gzhu.edu.cn/faculty/~csgjwang/ (Guojun Wang)

the potential to violate the user's privacy [3]. By collecting the user's requests, an untrustworthy LSP can infer personal information about the user, such as his/her location, preferences, mode of transport, and possibly his/her state of health [4]. Even worse, the LSP could disclose the user's private information to third parties for financial or other business advantage. Consequently, it is advantageous to protect the user's privacy.

Several approaches have been proposed to mitigate the threat of loss of privacy in LBSs. Most adopt an architecture based on a completely-trusted third party (CTP) [5, 6], which functions as an anonymizer. When a user issues a request, it is sent to the anonymizer to remove the user's caller identity before forwarding the request to the LSP. In addition to removing the caller identity, the anonymizer also blurs the location of the user, most often by "cloaking". Cloaking is where the user's actual location is replaced by a circle. The user is somewhere in this circle, as are $K-1$ other users, thereby providing K-anonymity [7, 8]. However, this degrades the POI service. Consequently, the POI results are returned to the anonymizer for refinement so that more accurate results can be returned to the user. While the user is likely to discard the results almost immediately, it is in the interests of the anonymizer to cache them in order to aid it to answer future requests. If other users can obtain their POI service directly from the anonymizer's cached results, it will not be required to send their requests to the LSP.

In Reference [9], we proposed a caching-based scheme to enhance user privacy for continuous LBSs; the scheme employs a two-level caching to cache result data. However, this scheme does not consider the user's neighbor cache and the anonymity based on user mobility. To further enhance user privacy and increase cache hit rate, we propose an enhanced privacy protection scheme based on caching and spatial K-anonymity (CSKA) for continuous LBSs, in this paper. Our scheme adopts a multi-level caching to cache users' results. When a user issues a query request in continuous LBSs, he/she first queries the results in the cache of the local client, user's neighbor, and anonymizer in turn. If the user can obtain the results directly from these caches, he/she will not be required to interact with the LSP, and it will not expose any information to the untrusted LSP. Otherwise, a spatial K-anonymity that includes K cells [10] is formed by the anonymizer, and it is sent to the LSP for query. In the process of forming a spatial K-anonymity, our scheme first utilizes the Markov model to predict the next query location according to the user mobility; then, it selects K cells based on the predicted location, cell's cache contribution rate, and data freshness to improve the cache hit rate, consequently reducing the number of the interactions between the user and LSP and minimizing the overhead of the LBS server.

The main contributions of this study are as follows:

(1) We propose a novel CSKA scheme to provide enhanced user privacy for continuous LBSs. Our scheme adopts multi-level caching to cache users' results; moreover, a few of the user queries can be answered directly by cache, which can reduce the risk of exposure of user information to untrusted LSP.

(2) We employ spatial K-anonymity to enhance user privacy. According to the cells that the user are required to query, the anonymizer selects K cells to form a cloaking region, which can conceal the real user's location.

(3) We utilize the Markov model to predict the users' next location according to the user's mobility. To improve the cache hit rate, the cloaking region will be formed according to the predicted location, cell's cache contribution rate, and data freshness, thus reducing the interaction between the users and LSP.

(4) We analyze the effectiveness and efficiency of our proposed CSKA scheme. The simulation results reveal that our proposed CSKA scheme can provide higher privacy protection than a few previous methods and that it can minimize the overhead of the LBS server.

The rest of this research paper is organized as follows: Section 2 presents the related work. Section 3 provides a general overview of our system model and the fundamental definition. Section 4 presents details of the proposed scheme. A detailed discussion on the security analysis is presented in Section 5, and an analysis of the system's performance-efficiency is presented clearly in Section 6. Finally, Section 7 concludes this paper.

2. Related work

In this section, we review certain available research on system architectures, the main techniques, and cache methods for privacy preservation in LBSs.

2.1. Architectures

There are mainly two types of commonly-used architectures of privacy preservation in LBSs: user-centralized architecture [11] and centralized architecture [12].

In the user-centralized architecture, users can communicate with an LSP directly. Prior to sending queries, the users are generally required to blur their location information. The typical methods include generating fake locations, adding noise, and forming anonymity. The advantage of this architecture is its convenient deployment. However, for a user requiring high privacy protection, these methods for blurring user location will substantially reduce his/her experience.

The centralized architecture introduces a trusted third-party component called anonymizer, which sits between the users and LSP. The main role of the anonymizer is to perform location anonymity so that the LSP cannot identify the real user from the other users, thereby protecting the privacy of the user. The introduction of the anonymizer improves the capability to protect user privacy. However, as all the submitted queries of users have to go through the anonymizer, it will be the performance bottleneck of this architecture.

2.2. Techniques

At present, certain techniques of location privacy protection have been proposed in LBSs. Generally, we classify them into two main categories: dummy-based techniques and generalized techniques.

In the dummy-based techniques, there are mainly fake location methods and pseudonym methods. The former primarily conceals the user's query by generating a few fake locations in order to reduce the risk of user privacy exposure [13]. For example, Zhang et al. [14] presented a method of generating fake locations uniformly according to the actual road network. The method is simple and incurs marginal computational overhead. However, it is challenging to effectively generate fake locations according to the actual situation to confound an attacker. The pseudonym methods conceal the user's true identity by using the user's temporary identity, so that the attacker cannot be associated with the real user. For example, Gong et al. [15] presented an incentive strategy to motivate users to change their pseudonyms according to the social relationship between mobile social groups. The user privacy-preserving degree of this method depends mainly on the association between the user's identity and the special location; therefore, the user needs to change his/her pseudonym frequently.

The generalized techniques ordinarily generalize a query point of a user into an area and then sends it to the LSP for query; thus, the attacker only knows that the user is in a specific region [16]. The typical method is K-anonymity, which forms a cloaking region containing at least K users; the attacker cannot distinguish a particular user from the other $K-1$ users. For example, Zhang et al. [17] presented a dual privacy-preserving scheme to protect user's trajectories and content privacy; here, the user can select different anonymizers to form the cloaking region. Zhao et al. [18] proposed a K-anonymity solution to prevent injection attacks against user location, in which the attacker cannot obtain knowledge about the user's fake location. In general, the generalized techniques increase the overhead of the LBS server because it provides inaccurate location queries to LSP. Furthermore, using only this technology in continuous LBSs can provide limited privacy protection.

2.3. Cache methods

In a few recent works, the cache methods have been applied to user privacy protection, and users can employ the cached data of previous queries to answer subsequent queries. For example, Amini et al. [19] designed a system architecture involving the pre-fetching of the service data to enhance user privacy; however, the mobile user needs to cache a large amount of data during the query. Shokri et al. [20] proposed a distributed privacy protection scheme based on a collaborative group in the LBS. Before a mobile user issues a query request to the LSP, he /she first queries his/her neighbors for service data to reduce the probability of location exposure. However, the solution does not pay attention to the design of the caching strategy to enhance user privacy. Park et al. [21] proposed a location-based cache strategy to support spatial query processing; furthermore, a mobile user pre-fetches data that are likely to be used in the near future. However, it is suitable only for wireless broadcast environments. Jung et al. [22] proposed a privacy-preserving scheme that employs a virtual individual server to preserve the location privacy in an LBS; this can prevent system performance degradation during anonymity through collaborative caching. Peng et al. [23] presented a privacy protection scheme that protects user trajectories through collaborative caching. It is challenging for an attacker to reconstruct the user's trajectory. Zhang et al. [24] presented a scheme to enhance user privacy protection through unified grid and cache. This scheme can prevent different users from querying an identical region; however, the overhead of the client is large. Niu et al. [25] presented a caching-based scheme in which two cache-based dummy location selection algorithms are designed to enhance user location privacy; however, this method is not well suited for continuous queries.

None of the above studies considered the user mobility in continuous LBSs. As a departure from these, we propose a method to form a cloaking region based on user mobility and adopt multi-level caching to improve user privacy.

3. The system model and definition

This section first describes the CSKA scheme to provide enhanced privacy in continuous LBSs. Then, we state the motivation for this work. Finally, we define the adversary model, privacy metric, and Markov model.

3.1. System architecture

To improve user privacy, we propose a CSKA scheme in continuous LBSs; its architecture is shown in Figure 1. It is a centralized architecture with three main entities: mobile clients, anonymizer, and LSP. Their main functions can be described as follows:

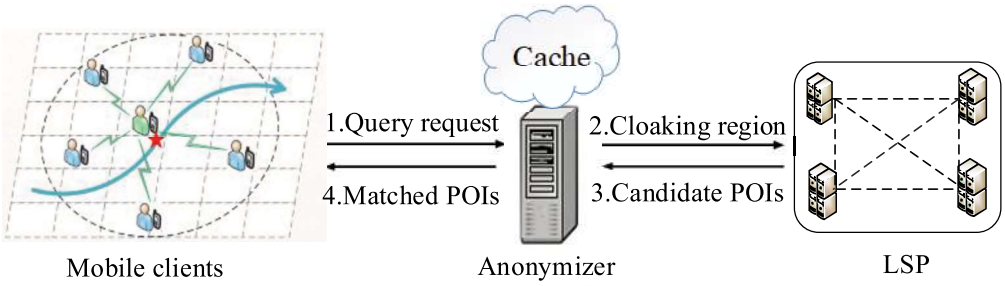


Figure 1: The architecture of CSKA

Mobile clients. Each mobile client carries a mobile device with rudimentary functions such as information processing, data storage, and global positioning (e.g., GPS); more importantly, it exhibits short-distance communication capabilities for it to communicate with other mobile users around it, through wireless communication protocols such as IEEE 802.11, Bluetooth, and Radio Frequency IDentification [26, 27]. Mobile clients can also collaborate to share query content in each other's cache.

Anonymizer. An anonymizer is a trusted entity between the mobile clients and LSP. Its main function is to form spatial K-anonymity and cache users' query results. It can also collect all the users' query information and count the number of queries issued in each cell in the past. In the process of continuous LBSs, an anonymizer can predict the user's next query according to the user mobility.

LSP. An LSP is an online location-based service solution provider with a number of LBS application servers that store map resources and location-related POIs such as restaurants, hotels, and supermarkets. It can provide users with timely location-based comprehensive query service. When receiving a user query request, it searches the database for the POIs that satisfies the user's needs according to the clocking region provided by the anonymizer and returns it to the user via the anonymizer.

The CSKA scheme adopts multi-level caching to reduce the risk of exposure of users' information to an LSP and combines spatial K-anonymity to enhance user privacy. The work process is as follows:

(1) When a user issues a query request in continuous LBSs, he/she first queries in the cache of the local client, user's neighbors, and anonymizer in turn. If the service data in the cache can satisfy the user's requirements, the user obtains the query results directly.

(2) Otherwise, the user's query request is sent to the anonymizer to form a spatial K-anonymity. We utilize the Markov model to predict the users' next location according to the user mobility and select K cells to form a cloaking region based on the predicted location, cell's cache contribution rate, and data freshness.

(3) Then, the cloaking region is sent to the LSP for query, and the LSP searches the POIs within the user query range from the database, obtains the query results, and returns them to the anonymizer.

(4) The anonymizer updates the query results to the cache and returns the matched user service data to the user. Similarly, the user also updates the cache and refines the matched service data to obtain accurate results.

3.2. Problem statement

In LBSs, users need to send information related to the current location to an untrusted LSP to obtain the desired service. We consider the following scenario: In densely populated areas, numerous mobile users need to send continuous range queries to an LSP that queries the same POIs, such as discounted information for nearby restaurants. There are two concerns: first, during continuous LBS queries, the submitted queries include the current locations and interests of the mobile user, and a user frequently submits these queries to the untrusted LSP; this increases the risk of identification of his/her trajectory by the LSP. Secondly, different users send the same query request to the LSP for repeated queries; this increases the risk of exposure of the user's privacy and the overhead of the LBS server. For the first issue, the general approach is to use K-anonymity; however, the selected dummy locations are vulnerable to data mining attacks aimed at identifying the real user. On the second question, we naturally consider the caching method, which permits users to obtain data directly from the cache; this enhances the user privacy and reduces the overhead of the LBS server. Nevertheless, improving the cache hit rate is a challenging problem.

To address the issues mentioned above, the fundamental concept of our proposed scheme is to integrate caching and spatial K-anonymity to provide enhanced privacy. Our solution employs a multi-level caching to cache the data previously queried by the user on the local client, user's neighbor, and anonymizer for subsequent queries to mitigate the risk of exposure of user information to an untrusted LSP. To improve the cache hit rate, we use Markov-based location prediction to predict the user's next location according to the user mobility in continuous LBS queries and select K cells to form a cloaking region according to the predicted location, cell's cache contribution rate, and data freshness. Simultaneously, in order to solve the risk of exploitation of the dummy location in K-anonymity, we adopt spatial K-anonymity to further enhance user privacy.

3.3. Adversary model and privacy metric

In our system, we assume that both the anonymizer and the mobile clients are trusted entities. The initiating query user and the neighboring users within the user's communication range will not collect the data information of the other party from the peer node and will not send malicious information to the other node. Additionally, the communication channel between them is secure, and available security schemes such as cryptographic and hashing algorithms can be employed to protect the integrity and confidentiality of the transmitted data [28, 29].

In our adversary model, we regard the main purpose of an adversary as the determination of the users' location associated with the true identity; the knowledge of the adversary is an important factor in assessing our scheme's privacy. Therefore, according to the background knowledge gained by the adversary, our scheme mainly considers two types of adversary models.

(1) The LSP in the system is regarded as an honest albeit curious entity that straightforwardly obtains all the user query information by monitoring the queries sent by the user; it also knows the privacy protection mechanisms used in the system. Based on this information, it can attempt to infer certain information related to the user's location, and it may even leak the user's sensitive information to a third party.

(2) An adversary may collect certain user query information by eavesdropping on the communication channel between the user and LSP and attempt to obtain certain sensitive

information about the user from the data, such as the user identity, sensitive query content, and location. It generally exhibits limited background knowledge of the user and can be localized and short-term operated.

In this paper, we quantify location privacy by entropy-based metrics [30], which is mainly determined by two factors. One is that the user obtains the privacy degree when he/she sends a query to the LSP. The other is the effect of caching, because certain users' query requests are answered by cached data. We will evaluate the privacy of our system mainly based on the cache hit rate; a higher cache hit rate indicates that fewer user queries are sent to the LSP, which can mitigate the risk of exposure of user information.

3.4. Markov model

In daily life, a user's mobility generally exhibits certain regularity, such as a fixed location that the user has to pass in turn on the way to and from work. In this study, we use the Markov model to predict the user's next query location according to the user mobility and select K cells to form spatial K-anonymity according to the predicted location; this can improve the cache hit rate of the user's next query.

The trajectory T of a moving object can be represented by a set of discrete two-tuple locations l at sampling time, which can be expressed as follows:

$$T = \{p_1,\ p_2,\ \cdots,\ p_n\} = \{(l_1,\ t_1),\ (l_2,\ t_2),\ \cdots,\ (l_n,\ t_n)\} \tag{1}$$

where p_i denotes the ith point of trajectory T and $p_1 \cdot t < p_2 \cdot t < \cdots < p_n \cdot t, (p_i \cdot t - p_{i-1} \cdot t < \varepsilon\ ,\ 2 \leq i \leq n)$. The location $p_i \cdot l$ can be denoted by a two-tuple (x, y), where x and y represent latitude and longitude, respectively.

A mobile user's trajectory can be expressed as an nth order Markov chain; the location l_i of each sampling point p_i on the trajectory is related only to the n positions $\{p_{i-n+1},\ p_{i-n+2},\ \ldots,\ p_{i-1},\ p_i\}$ ahead of it [31, 32].

$$\begin{aligned} &\Pr(p_{i+1} \cdot l = l_{i+1} | p_i \cdot l = l_i,\ p_{i-1} \cdot l = l_{i-1},\ \ldots,\ p_1 \cdot l = l_1) \\ &= \Pr(p_{i+1} \cdot l = l_{i+1} | p_i \cdot l = l_i,\ p_{i-1} \cdot l = l_{i-1},\ \ldots,\ p_{i-n+1} \cdot l = l_{i-n+1})\} \end{aligned} \tag{2}$$

where $p_i \cdot l = l_i$ denotes the location of sampling point p_i on the trajectory.

For the first n locations on the trajectory, the next location can be predicted by Equation (3):

$$\begin{aligned} &\Pr\ (p_{i+1} \cdot l = l_p | p_i \cdot l = l_i,\ p_{i-1} \cdot l = l_{i-1},\ \ldots,\ p_1 \cdot l = l_1) = \\ &max_{k=1}^{n} \Pr(p_{i+1} \cdot l = l_k | p_i \cdot l = l_i,\ p_{i-1} \cdot l = l_{i-1}, \ldots, p_{i-n+1} \cdot l = l_{i-n+1}) \end{aligned} \tag{3}$$

where l_p denotes the predicted next location on the trajectory. Therefore, there are n candidate locations on the user's trajectory. We can predict the next location by calculating the maximum probability of l_k.

4. Our proposed CSKA scheme

In this section, we first elaborate the process of LBS query in our proposed scheme. Then, we introduce the process of location prediction based on Markov. Finally, we present the method of cloaking region formation. The important notations used in our CSKA scheme are listed in Table I for reference.

Table 1: Summary of important symbols

Symbol	Description
E, E_n	Asymmetric and symmetric encryption functions
PK_S, SK_S	Public and private key pairs of LSP
PK_A, SK_A	Public and private key pairs of anonymizer
k_u	The usr's key
(c_i, r_j)	Cell identifier of POI
l_u	Location of the query point
D_u	Movement direction of the user
I_u	Cell identifier set that need to be queried
POI_type	Query content
MSG_{U2A}	The query request from user to anonymizer
MSG_{A2S}	Anonymizer forwards the user's request to LSP
MSG_{S2A}	LSP returns the candidate results to anonymizer
MSG_{A2U}	The matched results return from anonymizer to user

4.1. Process of LBS query

Our system first specifies a rectangular query area that is divided into $M \times M$ grid structure with the equal cell, and each cell has a unique identifier $(c_i, r_j), 1 \leq i, j \leq M$. If a user's coordinate is (x_i, y_i), its cell identifier (c_i, r_j) can be computed by Equation (4):

$$(c_i\ ,\ r_i) = \left(\left\lceil \frac{x_i - x_1}{(x_2 - x_1)/M} \right\rceil\ ,\ \left\lceil \frac{y_i - y_1}{(y_2 - y_1)/M} \right\rceil \right) \tag{4}$$

where (x_1, y_1) and (x_2, y_2) denote the bottom-left and top-right vertex coordinates, respectively, of the rectangular query area. Then, our system sets a desired threshold θ $(0 < \theta \leq 1)$ according to the tradeoff between service quality and privacy. In the process of searching for results in the cache, if the proportion of the matched cell identifiers in the cache is higher than θ, it implies that the cache can satisfy the user's query needs. That is, the user can obtain the query results directly from the cache without interacting with the LSP. The higher the value of θ is, the better the service quality will be, although it will reduce user privacy.

4.1.1. Query request

When a user issues a query request, he/she first identifies all the cell identifiers I_q that cover the query range of radius R; here, $I_q = \{(c_i, r_j)\}, 1 \leq i, j \leq M$. Then, the user searches for these identifiers I_q with identical POIs in the client cache, and the matched cell identifiers in the client cache is I_c. if $Num(I_c)/Num(I_q) \geq \theta$, and $Num()$ is a function for counting the number of cell identities, it implies that the client cache can satisfy the user's needs and that the user can obtain the result data directly. Otherwise, the user sends the POI type and the cell identifier set $I_u^1 = I_q - I_c$ that need to be queried to the cache of the neighbor clients for query. In mobile clients, we adopt the cooperative caching mechanism based on the signature augment tree structure proposed in reference [33] to achieve cache sharing among clients. Similarly, if the matched cell identifiers I_n from the neighboring users satisfy $(Num(I_c) + Num(I_n))/Num(I_q) \geq \theta$, the user obtains the result data directly. Otherwise, the user needs to send a request message MSG_{U2A} to the anonymizer.

$$MSG_{U2A} = E_{PK_A}\{l_u, D_u, I_u^2, \lambda, k_u, POI\_type\} \tag{5}$$

where l_u indicates the location of the user's current query point, D_u denotes the direction of user movement, I_u^2 indicates the cell identifier set that needs to be queried, and $I_u^2 = I_q - I_c - I_n$. λ indicates the minimum number of cell identifiers that the anonymizer requires to query, and $\lambda = \theta * Num(I_q) - Num(I_c) - Num(I_n)$. k_u indicates the randomly generated key of the user; and POI_type indicates the type of POI that the user queries, which is encrypted by asymmetric encryption E with the anonymizer's public key PK_A.

When receiving the request message MSG_{U2A}, the anonymizer first decrypts it with the private key SK_A. Then, the anonymizer matches each cell identifier of I_u^2 in the anonymizer's cache and obtains the matched cell identifier set I_a. If $Num(I_a) \geq \lambda$, the content of multi-level caches can satisfy user needs, and the user privacy can be protected through the caching because he/she does not send any query to the LSP. Otherwise, the anonymizer selects other $K - Num(I_u^3)$ cells based on the cell identifier set I_u^3 ($I_u^3 = I_q - I_c - I_n - I_a$) that needs to be queried to form a cloaking region. The detailed process of cloaking region formation is described in Section 4.3. Finally, the anonymizer forwards the user's request MSG_{A2S} to the LSP.

$$MSG_{A2S} = E_{PK_S}\{Region, POI\_type\} \tag{6}$$

where $Region$ indicates the cloaking region and POI_type indicates the query content. Then, the anonymizer encrypts the $Region$ and POI_type by asymmetric encryption E with the LSP's public key PK_S.

4.1.2. The results return

The LSP receives the MSG_{A2S}, decrypts it with the private key SK_S and obtains the K cells in the $Region$. Then, it searches the POIs according to the POI_type in these cells and obtains G POIs. If the location of the jth POI is $(x_j, y_j)(1 \leq i, j \leq G)$, we can obtain the corresponding cell identifier for each POI by Equation (4) and thus obtain the set $\varphi_r(1 \leq r \leq K)$ of cell identifiers containing these POIs. Finally, the LSP encrypts the φ_r by asymmetric encryption E with the anonymizer's public key PK_A, forms the candidate result set message MSG_{S2A}, and returns it to the anonymizer.

$$MSG_{S2A} = E_{PK_A}\{\varphi_r\} \quad (1 \leq r \leq K) \tag{7}$$

The anonymizer receives the MSG_{S2A}; it first decrypts it with the private key SK_A and updates these POIs and corresponding cell identifiers to its cache. Then, the anonymizer matches the cell identifiers by comparing $\varphi_r(1 \leq r \leq K)$ with the I_u^3 previously needed be queried and obtains all the cell identifier set I_u^2 along with the POIs that the user previously needs to query. Finally, the anonymizer encrypts the I_u^2 along with the POIs by symmetric encryption En with the key k_u. The query result message MSG_{A2U} that the anonymizer forwards to the user is

$$MSG_{A2U} = En_{k_u}\{I_u^2\ ,\ (x_i\ ,\ y_j)\} \tag{8}$$

The user receives the MSG_{A2U}; he/she first decrypts it with the key k_u and updates it to the client cache. When the client cache cannot satisfy the storage requirements, we need to manage the cell identifiers along with the POIs in caching. In our scheme, we adopt the cache replacement strategy based on the user's movement trend. According to the user's current location and movement direction, we replace the cell identifiers together with the POIs that are farthest from the predicted location with the opposite movement direction; this can increase the cache hit rate. Finally, the user computes the POIs contained in the query range of radius R and obtains accurate results.

4.2. Location prediction based on Markov

In our CSKA scheme, we utilize the Markov model to predict the next location of a mobile user according to the user mobility and form a spatial K-anonymity based on the predicted location to enhance privacy. We first obtain certain stay points of significance from the users' historical trajectories. Then, we construct a Markov state-transit matrix comprising the probabilities of users moving between locations. Finally, we can predict the future states according to this matrix and the user's current state.

4.2.1. The stay points

In this section, we extract the stay points from the users' historical trajectories in the anonymizer. Suppose the historical trajectory of user u_i is represented by $T^i = \{p_1^i,\ p_2^i,\ \cdots,\ p_n^i\}$. Consider $p_j^i = (l_j^i,\ t_j^i) = (x_j^i,\ y_j^i,\ t_j^i)$, where x_j^i and y_j^i are the geographical coordinates of the jth position in the trajectory of user u_i and t_j^i is the timestamp when the user is at the jth position. We can calculate the distance between the two positions p_a^i and p_b^i with Equation (9), and $1 \leq a, b \leq n$:

$$Dis(a,b) = 2R \cdot arcsin\sqrt{X + Ycos(p_a \cdot x)cos(p_b \cdot x)} \tag{9}$$

where R is the radius of the earth, $X = haversin(p_a \cdot x - p_b \cdot x)$, $Y = haversin(p_a \cdot y - p_b \cdot y)$, and x and y are the longitude and latitude of the moving position p_a and p_b.

A stay point is the location at which a mobile user stays in the area within the distance ΔD_r for at least Δt_r time; here, ΔD_r and Δt_r are the thresholds of the distance and time, respectively. In our paper, the stay point is defined as a set of continuous positions that satisfy the following conditions:

$$T = (p_m\ ,\ p_{m+1},\ \cdots,\ p_n) \tag{10}$$

For $\forall m < i < n,\ Dis\,(p_m, p_i) \leq \Delta D_r,\ Dis\,(p_m, p_{n+1}) > \Delta D_r$, and $|p_n \cdot t - p_m \cdot t| \geq \Delta t_r$. We use the center of T to represent the coordinate of the stay point sp, which can be represented by a four tuple:

$$sp = (\bar{x},\ \bar{y}\ ,\ t_i\ ,\ t_o) \tag{11}$$

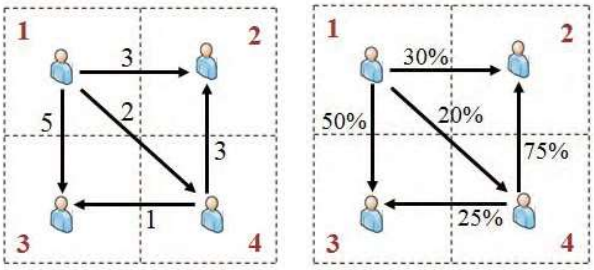
$$sp \cdot \bar{x} = \frac{\sum_{i=m}^{n} p_i \cdot x}{|p|} \tag{12}$$

$$sp \cdot \bar{y} = \frac{\sum_{i=m}^{n} p_i \cdot y}{|p|} \tag{13}$$

where t_i and t_o indicate the time at which the mobile user enters and leaves the stay point, respectively, and $|p|$ indicates the number of continuous positions that the stay point contains. Then, we can extract all sets of stay points SP^i of user u_i from his/her historical trajectories, $SP^i = \{sp_1^i\ ,\ sp_2^i\ ,\ \ldots\ ,\ sp_n^i\}$.

4.2.2. Markov chain

The Markov chain is used to describe the probabilities that a user moves from one location to another. We map the stay points of the users' historical trajectories to the cell identifiers by using the coordinates of the stay points, and the user trajectory can be represented by a sequence of the cell identifiers. Then, we construct a weighted graph $G(V, E, W)$; V indicates



(a) Weight (b) Probability

Figure 2: The Markov state transition.

the set of cell identifiers, E indicates a set of edges from one cell identifier to another, and W indicates the weight of the edge. The weight $w_{i\to j}$ of the edge (i, j) is defined as the number of movements of the user from position i to j. Finally, we can calculate the probability $p_{i,j}$ that a user moves from position i to j as follows:

$$p_{i,j} = \frac{w_{i\to j}}{\sum_{j=1}^{M^2} w_{i\to j}} \tag{14}$$

$$\sum_{j=1}^{M^2} p_{i,j} = 1 \tag{15}$$

As shown in Figure 2, a one-step state transition of the Markov chain occurs, and $p_{1,3} = 5/(5+3+2) = 50\%$. After counting all the transmissions on all the cells, we can obtain the Markov state-transit matrix $Pr = \{p_{1,j}, p_{2,j}, ..., p_{M,j}\}$, $1 \le j \le M$.

4.2.3. The location prediction

According to the Markov state-transit matrix Pr and the user's current location l_j^i, we can obtain the next location l_p^i by Algorithm 1 and can obtain the cell identifier of the predicted location l_p^i.

Algorithm 1 Location prediction based on Markov

Input: The user's historical trajectories T and the current location l_j^i
Output: The next location l_p^i

1: Get the stay points $SP^i = SP\_Detection(T)$;
2: Mapping SP^i to $M \times M$ grid structure;
3: Constructing a weighted graph $G(V, E, W)$;
4: Constructing a Markov state-transit matrix Pr;
5: Calculate the state transition probability according to the current location l_j^i, and the maximum probability is l_p^i;
6: **return** l_p^i

4.3. Forming cloaking region

In the process of forming a cloaking region, we select K cells to satisfy spatial K-anonymity according to the user's predicted location, cell's cache contribution rate, and data freshness to improve the cache hit rate.

4.3.1. Cache contribution rate

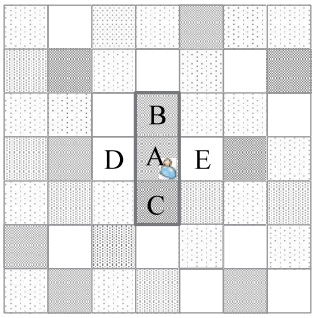


Figure 3: The cell selection for forming cloaking region

As shown in Figure 3, this is a 7×7 grid structure with the equal cell, the density of points in the cell represents the probability that the user will issue a query. The blank cells indicate that these regions have almost no users' queries and are likely to have swamps, rugged mountains, or lakes on it. The user who needs to issue the query is in the cell A, and we need to select the other two cells to form a cloaking region. If we select the cells D and E, the attacker can conveniently infer the user's true cell by related background knowledge. To improve the cache contribution rate, we need to select other cells around the user with the highest query probability to form the clocking region. Therefore, we should select the cells B and C.

Suppose P_i represents the probability that these users send queries in each cell, the cache contribution rate of the $K - Num(I_u^3)$ cells can be calculated by $\sum_{i=1}^{K-Num(I_u^3)} P_i$, and $\sum_{i=1}^{M^2} P_i = 1$. When we select other $K - Num(I_u^3)$ cells to form the cloaking region, the cells should have the largest cache contribution rate C_s:

$$C_s = \max \sum_{i=1}^{K-Num(I_u^3)} P_i \tag{16}$$

4.3.2. Data freshness

The cached data of a cell has a certain lifetime. Before the cached data expires, we will update it, particularly for those cells that exhibit a high query probability. The data freshness of the cell can be expressed as

$$\varphi = 1 - \sqrt{t^2/T^2}, T \geq t \tag{17}$$

where T denotes the lifetime of the cached data and t denotes the time that the data has been cached. When a cloaking region is formed, the average data freshness of the selected $K - Num(I_u^3)$ cells can be expressed as

$$\bar{\varphi} = \frac{1}{K - Num(I_u^3)} \sum_{i=1}^{K-Num(I_u^3)} \left(1 - \sqrt{t_i^2/T_i^2}\right) \tag{18}$$

4.3.3. The cloaking region

Considering the above two factors, we select the cells that exhibit the maximum cache contribution rate and the minimum data freshness to form a cloaking region; the selected cell set can be represented as

$$C_d = \max \sum_{i=1}^{K-Num(I_u^3)} P_i \cdot \frac{\sum_{i=1}^{K-Num(I_u^3)} \sqrt{t_i^2/T_i^2}}{K - Num(I_u^3)} \tag{19}$$

Algorithm 2 illustrates the process of forming a cloaking region. We first count the number of cells that need to be queried and select N cells around the predicted location l_p. Simultaneously, we select the $2K$ cells with the highest query probability from this N cells. Then, we randomly select W of the $2K$ cells as the candidate set C_S, and the cache contribution rate and the data freshness of each candidate set are calculated. Finally, we obtain the maximum value C_d of C_S as the cell set selected in the cloaking region.

Algorithm 2 Process of forming cloaking region

Input: K (Anonymity degree), l_p^i (The predicted location of the user u_i), I_u^3(Cell set that needs to be queried), N (System parameter and $2K < N$)
Output: *Region* (The cloaking region)

The number of cells that need to be selected is $W = K - Num(I_u^3)$;
Select N cells around the predicted location l_p^i;
Sort N cells in descending order by P_i, and select the first $2K$ cells;
Randomly select W out of $2K$ cells as the candidate set C_S;
$C = \{C' | C' \subset C_S, |C'| = W\}$;
for each $C \in C'$ **do**
    $C_{di} = \sum_{i=1}^{W} P_i \cdot \frac{1}{W} \sum_{i=1}^{W} \sqrt{t^2/T^2}$;
end for
$C_d = \max \sum_{i=1}^{|C_s|} C_{di}$;
$Region = I_u^3$;
for each $(c_i, r_i) \in C_d$ **do**
    $Region = Region + (c_i\ ,\ r_j)$;
end for
return *Region*

5. Security analysis

In the next section, we perform a security analysis of the CSKA scheme to demonstrate how it can enhance user privacy. According to the adversary model defined in Section 3.3, the security of the proposed scheme is analyzed to demonstrate how it can protect against dishonest LSPs and/or eavesdropping attacks.

5.1. Privacy against LSP

When a user makes a query request, he/she first searches the results in the cache of the user, user's neighbor, and anonymizer in turn. If the cached data can satisfy the user's need, the user can obtain the result data directly. During this process, the user does not send any query to the untrusted LSP; therefore, the LSP cannot obtain any information from the user.

If the user cannot obtain the result that satisfies the user's needs in the cache, we need to form a cloaking region on the anonymizer; it is forwarded to the LSP by the anonymizer. However, the request message MSG_{A2S} sent to the LSP contains only the query content *POI_type* and the cloaking region *Region*. From this information, the LSP only knows the user's *POI_type*, which cannot be associated with a specific user, and it does not obtain the exact location of the user.

In the process of forming a cloaking region, we select K cells to satisfy spatial K-anonymity based on the user's predicted location, cell's cache contribution rate, and data freshness. The cloaking region contains K cells, which do not necessarily contain the user's real location because the cell containing the user's real location may already have the result data in the cache. Even if the real user's location is in one of the K cells, we select other cells with the highest query probability as the cloaking region, and each cell contains certain users; therefore, the LSP can speculate that the probability of a specific user is at most $1/K$. Although the LSP knows the query probability P_i $(1 \leq i \leq M^2)$ of each cell in the grid structure, these cells are not focused on a sensitive area, and the LSP cannot successfully estimate the real location from the submitted cloaking region.

From the above analysis, we observe that our CSKA scheme can effectively resist the inference attack of the LSP.

5.2. Resistance to eavesdropping attacks

In the CSKA scheme, we employ cryptographic techniques to handle eavesdropping attacks using asymmetric and symmetric encryption to encrypt all the messages transmitted between the user and LSP.

When messages are sent to the anonymizer (either the request from the user or the results from the LSP), they are encrypted by asymmetric encryption using the public key of the anonymizer. Because an attacker will not have access to the anonymizer's private key, it cannot decrypt these messages.

When the anonymizer forwards the request message to the LSP, it is asymmetrically encrypted using the public key of the LSP. Because an attacker does not have access to the private key of the LSP, it cannot decrypt the request. The results message from the anonymizer to the user is symmetrically encrypted with a secret key known only to the anonymizer and user. Because an attacker does not have access to this, it cannot decrypt the results.

6. Evaluation

In this section, we evaluate the effectiveness and efficiency of our proposed CSKA scheme and conduct our experiments focusing on the cache hit rate, system performance, and overhead of the LBS server.

6.1. Simulation setup

In our simulation, we deployed 20,000 mobile users on a 10 km $\times$ 10 km Beijing map; all of them followed a walking mobility model [34, 35]. The local map is divided into $M \times M$ cells, and the user query probability for each cell can be obtained through the Google Maps API. The user can issue successive LBS queries during the move, and we set ΔD_r= 400 m and Δt_r= 30 min to obtain a few stay points. We randomly generated 10,000 POIs on the map, and they were set up as restaurants and hotels. The user can query the cached data of neighboring users within 50 m. The main experimental parameters are shown in Table II.

Table 2: Experimental Parameter Settings

Parameter	Values
The number of mobile users	20,000
Bottom-left vertex (x_1, y_1)	$(0, 0)$km
Top-right vertex (x_2, y_2)	$(10, 10)$km
The query range radius R	0.5km
The number of POIs	10,000
Continuous query points	1-20
Anonymity degree K	10-50
The number of cells	1,000-10,000
The threshold value	0.1–1.0

We carry out all the experiments on a PC with an Intel(R) Core(TM)i7-6700 CPU @ 3.40 GHz 3.41 GHz and 8 GB RAM; the CSKA schemes were implemented with the development environment of Java Development Kit and MyEclipse. We select the MobiCrowd [20] and CaDSA [25] schemes as the baseline algorithms. For continuous LBS queries, we measured the average processing times on the mobile clients, anonymizer, and LSP as well as the average communication cost between a user and the neighbors/anonymizer and that between the anonymizer and LSP.

6.2. Effectiveness analysis

The effectiveness of our CSKA scheme is mainly analyzed by changing the continuous query points n, anonymity degree K, and number of cells M.

6.2.1. The matching of the cells

When K=30, $M = 5000$, and $\theta = 0.8$, a user needs to query the POIs in the cells that are covered in the query range of radius R. When the user sends 10 continuous LBS queries, we analyze the proportion of matched cells in the local client, LSP, neighbors, and anonymizer. As observed in the Figure 4, when the number of user's queries increases, the number of matched cells obtained from the local client, neighbors, and anonymoizer gradually increases, whereas the number of matched cells obtained from the LSP gradually reduces; eventually, they tend to be in a relatively stable state.

This is because when the user initially queries, the local client cache has almost no required POIs, and the user can only obtain result data from the surrounding neighbors, anonymizer, and LSP. In this query process, we utilize the Markov model to predict the next query location according to the user mobility and select K cells based on the predicted location, cell's cache contribution rate, and data freshness to form a spatial K-anonymity. Certain cells of the future query has been previously queried by the spatial K-anonymity, and these result data are stored in the cache of the mobile clients or anonymizer. Therefore, as the number of user queries increases, the query results obtained by the user from the local client, neighbors, and anonymizer increases, and only a small number of unmatched cells need to be queried by the LSP.

If the user can directly obtain the result data from the cache of the local client, the user's privacy is higher, and the system overhead is smaller. On the contrary, if the user obtains more result data from the LSP, the user's privacy reduces and the system overhead increases. As shown in this figure, when the number of queries increases, the proportion of the matched

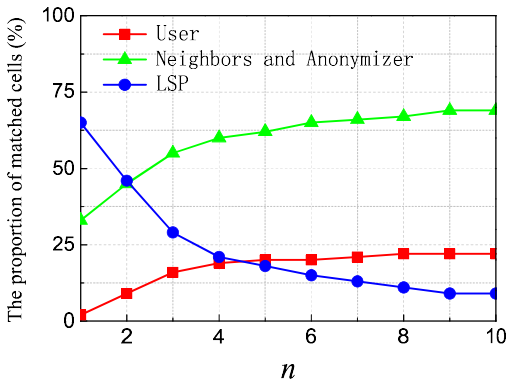


Figure 4: Result data obtained from different entities

Table 3: Comparison of result data obtained from different entities

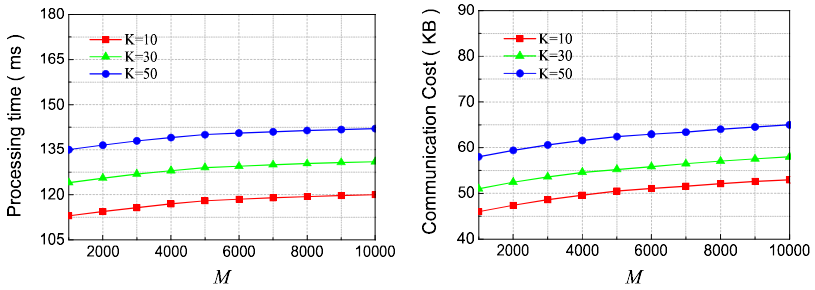
Different entities	Privacy degree	System overhead	Proportions of matched cells
Local client	High	Low	22%
Neighbors and anonymizer	Medium	Medium	69%
LSP	Low	High	9%

cells by the user from the local client, neighbors and anonymizer, and LSP is 22%, 69% and 9%, respectively. The comparison of the result data obtained from the different entities is presented in Table 3.

6.2.2. Effect of number of cells M.

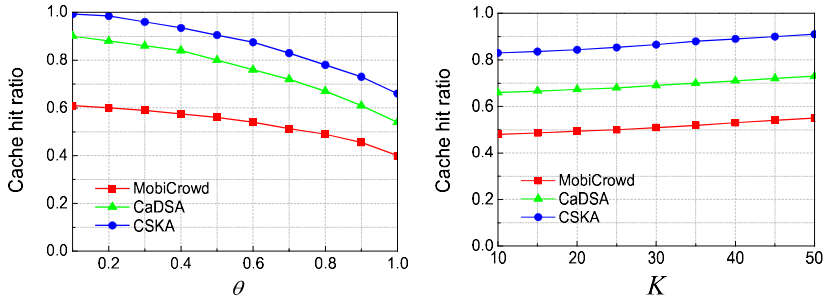
In the case of $\theta = 0.8$, $n = 10$ and $K = 10$, and 30 and 50, we evaluate the impact of the parameters M and K on the system performance. As observed in the Figure 5, both the communication cost and the processing time increase with the increase in M. Moreover, the increase in the parameter K value also increases the system overhead.

This is because the number of cells M increases and the user needs to match more cell identifiers within the specified query range, which increases the overhead for processing these cell identifiers. Similarly, for a larger K, the cloaking region that needs to be queried is larger, thus increasing the system overhead. Therefore, the system overhead of user queries increases with the increase in $M$and K.



(a) Processing time. (b) Communication cost.

Figure 5: Effect of number of cells M.



(a) The effect of the θ vs. $K = 30$. (b) The effect of the k vs. $\theta = 0.8$.

Figure 6: The comparison of the cache hit rate.

6.3. Comparison

We mainly compare our CSKA scheme with the CaDSA and MobiCrowd methods in terms of cache hit rate and LBS server overhead to illustrate the effectiveness of our proposed scheme.

6.3.1. Cache hit rate.

When $M = 5000$ and $n = 10$, we compare the cache hit rate of the CaDSA, MobiCrowd, and CSKA methods under varying θ and K.

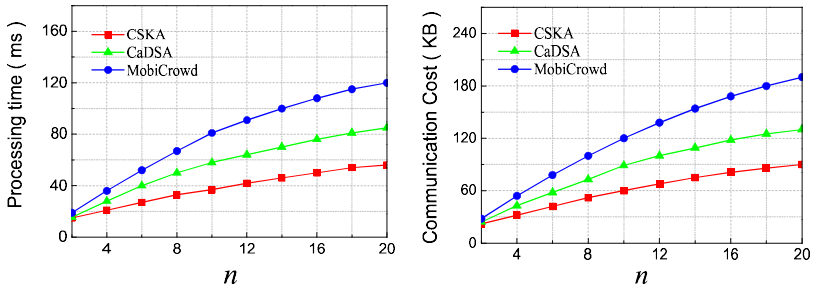
As shown in Figure 6(a), the cache hit rate of all the three methods decreased with the increase in θ at the $K = 30$; moreover, this rate for CSKA is higher than those for CaDSA and MobiCrowd. This is because with the increase in θ, the user needs to match more cell identifiers in the cache to satisfy the user's query requirements, and the cache hit rate will be reduced; however it can yield better service quality. Furthermore, the CSKA scheme is mainly based on the user's mobility to form the cloaking region; therefore, the CSKA exhibits a higher cache hit rate relative to the MobiCrowd and CaDSA methods.

As observed from Figure 6(b), the cache hit rate of all the three methods increased with K at $\theta = 0.8$; moreover, the rate for CSKA is higher than those for CaDSA and MobiCrowd. This is because the anonymizer forms a larger cloaking region as K increases, and LSP returns more query results to the cache of the anonymizer and mobile clients to satisfy the users' query requirements; then, the cache hit rate increases accordingly.

6.3.2. The Overheard of the LBS server.

When $M = 5000$, $\theta = 0.8$, and $K = 30$, we compare the overhead of the CSKA, CaDSA, and MobiCrowd methods on the LBS server under varying n.

As observed from Figure 7, with the increase in n, the advantage of the CSKA method relative to the CaDSA and MobiCrowd methods is higher in terms of the processing time and communication cost of the LBS server. This is because the CSKA method selects K cells to form a cloaking region according to the predicted location, cell's cache contribution rate, and data freshness, thus effectively improving the cache hit ratio and reducing the number of user information queries in the LBS server. However, the MobiCrowd simply uses the cache, and the CaDSA considers only the user's query probability when forming a cloaking region. Therefore, the CSKA scheme has substantial advantages over the CaDSA and MobiCrowd methods in terms of the LBS server overhead.



(a) Processing time. (b) Communication cost.

Figure 7: The comparison of the LBS server overhead.

7. Conclusion

In this paper, we proposed a CSKA scheme which adopts multi-level caching and spatial K-anonymity to provide enhanced user privacy for continuous LBSs. When forming a spatial K-anonymity, our scheme first utilizes the Markov model to predict the next query location according to the user's mobility and then selects K cells based on the predicted location, cell's cache contribution rate, and data freshness to improve cache hit rate; this reduces the interaction between users and LSP and improves user privacy. Security analysis reveals that the CSKA scheme can effectively protect user privacy and that it can counter eavesdropping attacks and LSP inference attacks. The simulation results also verify that our proposed scheme exhibits higher privacy protection than the previous solutions.

In the process of forming spatial K-anonymity, we only consider selecting the cells to form a cloaking region based on the prediction location of a single user. However, we did not consider the case of multiple user queries. Therefore, in the future work, we will further consider the selection of appropriate cells based on multiple users' predicted locations to form a cloaking region when multiple users simultaneously issue queries so that they can achieve better privacy protection. This will also be a multi-objective optimization problem. Additionally, the security of information sharing between users and their neighbors is also an important issue worth studying.

References

[1] X. Wang, A. Pande, J. Zhu, P. Mohapatra, Stamp: enabling privacy-preserving location proofs for mobile users, IEEE/ACM Transactions on Networking 24 (2016) 3276–3289.

[2] Q. Liu, G. Wang, F. Li, S. Yang, J. Wu, Preserving privacy with probabilistic indistinguishability in weighted social networks, IEEE Transactions on Parallel & Distributed Systems 28 (2017) 1417–1429.

[3] S. Zhang, G. Wang, Q. Liu, J. H. Abawajy, A trajectory privacy-preserving scheme based on query exchange in mobile social networks, Soft Computing 22 (2018) 6121–6133.

[4] Y. Sun, M. Chen, L. Hu, Y. Qian, M. M. Hassan, Asa: Against statistical attacks for privacy-aware users in location based service, Future Generation Computer Systems 70 (2017) 48–58.

[5] B. Gedik, L. Liu, Protecting location privacy with personalized k-anonymity: Architecture and algorithms, IEEE Transactions on Mobile Computing 7 (2008) 1–18.

[6] R. Schlegel, C.-Y. Chow, Q. Huang, D. S. Wong, User-defined privacy grid system for continuous location-based services, IEEE Transactions on Mobile Computing 14 (2015) 2158–2172.

[7] S. Zhang, Q. Liu, Y. Lin, Anonymizing popularity in online social networks with full utility, Future Generation Computer Systems 72 (2016).

[8] Y. Zhang, W. Tong, S. Zhong, On designing satisfaction-ratio-aware truthful incentive mechanisms for k-anonymity location privacy, IEEE Transactions on Information Forensics and Security 11 (2016) 2528–2541.

[9] S. Zhang, Q. Liu, G. Wang, A caching-based privacy-preserving scheme for continuous location-based services, in: The 9th International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage (SpaCCS), Springer International Publishing, 2016, pp. 73–82.

[10] G. Ghinita, K. Zhao, D. Papadias, P. Kalnis, A reciprocal framework for spatial k-anonymity, Information Systems 35 (2010) 299–314.

[11] R.-H. Hwang, Y.-L. Hsueh, J.-J. Wu, F.-H. Huang, Socialhide: A generic distributed framework for location privacy protection, Journal of Network and Computer Applications 76 (2016) 87–100.

[12] R.-H. Hwang, Y.-L. Hsueh, H.-W. Chung, A novel time-obfuscated algorithm for trajectory privacy protection, IEEE Transactions on Services Computing 7 (2014) 126–139.

[13] S. Gao, J. Ma, W. Shi, G. Zhan, C. Sun, Trpf: A trajectory privacy-preserving framework for participatory sensing, IEEE Transactions on Information Forensics and Security 8 (2013) 874–887.

[14] S. Zhang, G. Wang, Q. Liu, X. Wen, J. Liao, A trajectory privacy-preserving scheme based on dual-k mechanism for continuous location-based services, in: 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), IEEE, 2017, pp. 1004–1010.

[15] X. Gong, X. Chen, K. Xing, D.-H. Shin, M. Zhang, J. Zhang, Personalized location privacy in mobile networks: A social group utility approach, in: 2015 IEEE Conference on Computer Communications (INFOCOM), IEEE, 2015, pp. 1008–1016.

[16] G. Natesan, J. Liu, An adaptive learning model for k-anonymity location privacy protection, in: 2015 IEEE 39th Annual Computer Software and Applications Conference (COMPSAC), IEEE, 2015, pp. 10–16.

[17] S. Zhang, G. Wang, M. Z. A. Bhuiyan, Q. Liu, A dual privacy preserving scheme in continuous location-based services, IEEE Internet of Things Journal (2018). Doi:10.1109/JIOT.2018.2842470.

[18] P. Zhao, J. Li, F. Zeng, F. Xiao, C. Wang, H. Jiang, Illia: Enabling k-anonymity-based privacy preserving against location injection attacks in continuous lbs queries, IEEE Internet of Things Journal 5 (2018) 1033–1042.

[19] S. Amini, J. Lindqvist, J. Hong, J. Lin, E. Toch, N. Sadeh, Caché: caching location-enhanced content to improve user privacy, in: Proceedings of the 9th international conference on Mobile systems, applications, and services, ACM, 2011, pp. 197–210.

[20] R. Shokri, G. Theodorakopoulos, P. Papadimitratos, E. Kazemi, J.-P. Hubaux, Hiding in the mobile crowd: Locationprivacy through collaboration, IEEE Transactions on Dependable and Secure Computing 11 (2014) 266–279.

[21] K. Park, Y.-S. Jeong, A caching strategy for spatial queries in mobile networks., Journal of Information Science and Engineering 30 (2014) 1187–1207.

[22] K. Jung, S. Jo, S. Park, A game theoretic approach for collaborative caching techniques in privacy preserving location-based services, in: 2015 International Conference on Big Data and Smart Computing (BigComp), IEEE, 2015, pp. 59–62.

[23] T. Peng, Q. Liu, D. Meng, G. Wang, Collaborative trajectory privacy preserving scheme in location-based services, Information Sciences 387 (2017) 165–179.

[24] S. Zhang, K.-K. R. Choo, Q. Liu, G. Wang, Enhancing privacy through uniform grid and caching in location-based services, Future Generation Computer Systems (2017).

[25] B. Niu, Q. Li, X. Zhu, G. Cao, H. Li, Enhancing privacy through caching in location-based services, in: 2015 IEEE Conference on Computer Communications (INFOCOM), IEEE, 2015, pp. 1017–1025.

[26] A. Bhaskar, M. Qu, E. Chung, Bluetooth vehicle trajectory by fusing bluetooth and loops: Motorway travel time statistics, IEEE Transactions on Intelligent Transportation Systems 16 (2015) 113–122.

[27] M. Abdel-Basset, G. Manogaran, M. Mohamed, Internet of things (iot) and its impact on supply chain: A framework for building smart, secure and efficient systems, Future Generation Computer Systems 86 (2018) 614–628.

[28] X. Li, M. H. Ibrahim, S. Kumari, A. K. Sangaiah, V. Gupta, K. K. R. Choo, Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks, Computer Networks 129 (2017) 429–443.

[29] G. Wang, Q. Du, W. Zhou, Q. Liu, A scalable encryption scheme for multi-privileged group communications, The Journal of Supercomputing 64 (2013) 1075–1091.

[30] A. Serjantov, G. Danezis, Towards an information theoretic metric for anonymity, in: International Workshop on Privacy Enhancing Technologies, Springer, 2002, pp. 41–53.

[31] M. Chen, Y. Liu, X. Yu, Nlpmm: A next location predictor with markov modeling, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2014, pp. 186–197.

[32] M. Chen, W. Li, X. Chen, Z. Li, S. Lu, D. Chen, Lpps: A distributed cache pushing based k -anonymity location privacy preserving scheme, Mobile Information Systems,2016,(2016-7-17) 2016 (2016) 1–16.

[33] Q. Zhu, D. L. Lee, W. C. Lee, Collaborative caching for spatial queries in mobile p2p networks, in: IEEE International Conference on Data Engineering, 2011, pp. 279–290.

[34] K. Lee, S. Hong, S. J. Kim, I. Rhee, Slaw: A new mobility model for human walks, in: The 28th Conference on Computer Communications (INFOCOM), 2009, pp. 855–863.

[35] S. Yang, X. Yang, C. Zhang, E. Spyrou, Using social network theory for modeling human mobility, IEEE Network 24 (2010) 6–13.

Shaobo Zhang received the B.Sc. and M.Sc. degree in computer science both from Hunan University of Science and Technology, Xiangtan, China, in 2003 and 2009 respectively, and the Ph.D. degree in computer science from Central South University, Changsha, China, in 2017. He is currently a Lecture at School of Computer Science and Engineering of the Hunan University of Science and Technology, China. His research interests include privacy and security issues in social networks and cloud computing.

Xiong Li received the master's degree in mathematics and cryptography from Shaanxi Normal University, China in 2009 and Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications, China in 2012. He is currently an Associate Professor at School of Computer Science and Engineering of the Hunan University of Science and Technology, China. He has published more than 70 referred journal papers in his research interests, which include cryptography, information security, cloud computing security etc. He is currently an Editor of Telecommunication Systems and KSII Transactions on Internet and Information Systems. He has served on TPC member of several international conferences on information security and reviewer for more than 30 ISI indexed journals. He is a receipt of Journal of Network and Computer Applications 2015 best research paper award.

Zhiyuan Tan (M'13) received the B.Eng. degree in computer science and technology from Northeastern University, China, the M.Eng. degree in software engineering from the Beijing University of Technology, China, and the Ph.D. degree in computer systems from the University of Technology Sydney, Ultimo, NSW, Australia. He was a Post-Doctoral Researcher of cybersecurity with the University of Twente, The Netherlands, from 2014 to 2016; a Research Associate with the University of Technology Sydney, in 2014; and a Senior Research Assistant with La Trobe University, Australia, in 2013. He is currently a Lecture of cybersecurity with the School of Computing, Edinburgh Napier University, U.K. His research interests include cybersecurity, machine learning, pattern recognition, data analytics, virtualization, and cyber-physical system. He is an EAI and BCS Member.

He was the Chair of international workshops and conferences, such as SECSOC, SITN, EAI Future 5V, and EAI BD:TA 2018. He serves on the Editorial Board for the International Journal of Computer Sciences and its Applications. He is an Associate Editor of the IEEE Access and an Organizer of Special Issues for the Ad Hoc and Sensor Wireless Networks Journal, the International Journal of Distributed Sensor Networks, the Computers and Electrical Engineering, and the IEEE Access

Tao Peng received the B.Sc. degree in computer science from Xiangtan University, Xiangtan, China, in 2004, the M.Sc. degree in circuits and systems from Hunan Normal University, Changsha, China, in 2007, and the Ph.D. degree in computer science from Central South University, Changsha, China, in 2017. She is a lecture in the School of Computer Science and Technology, Guangzhou University, China. Her research interests include network and information security issues.

Guojun Wang received B.Sc. degree in Geophysics, M.Sc. degree in Computer Science, and Ph.D. degree in Computer Science, at Central South University, China, in 1992, 1996, 2002, respectively. He is a Pearl River Scholar Professor of Higher Education in Guangdong Province, a Doctoral Supervisor of School of Computer Science and Technology, Guangzhou University, China. He had been a Professor at Central South University, China; an Adjunct Professor at Temple University, USA; a Visiting Scholar at Florida Atlantic University, USA; a Visiting Researcher at the University of Aizu, Japan; and a Research Fellow at the Hong Kong Polytechnic University, HK. His research interests include cloud computing, mobile computing, trustworthy/dependable computing, cyberspace security, recommendation systems and mobile healthcare systems. He is a Senior Member of CCF and a member of IEEE, ACM, and IEICE.

Shaobo Zhang

Xiong Li

Zhiyuan Tan

Tao Peng

Guojun Wang