A Privacy Preserving Location Based Service System

Xinxin Zhao, Huiji Gao, Lingjun Li, Huan Liu, Guoliang Xue
Arizona State University

***Abstract*—Location based service is an indispensable part of today's mobile era. While it brings a lot of benefits to people, the breach to individual location privacy is always a concern and impedes the smooth development of location based service. A user can be easily tracked once she subscribes or uses the service from an untrusted location server. In this paper, we try to address this problem by proposing a secure and efficient location based service system. In our system, a user does not leak any of her location information while she can still obtain the desired information associated with the location. We propose a novel method to map a user's current location to the index of the information stored in the location server. We demonstrated the efficiency of our system through simulations.**

I. Introduction

Smart phone market has grown very fast in the past decade. In addition to traditional functions, such as texting and phone calling, a powerful computing capability and high portability grant smart phones many other functions. Among those novel functions, location based service (LBS) is one of the most popular and important ones. This service makes smart phones smart by letting phones sense the environment change and make response to the change.

Our work is motivated by one of the typical applications of LBS — Points of Interest (POIs) recommendation. Specifically, when a user stays at a new location and makes a query regarding the location point to a LBS server, maintained by some LBS provider, the server returns all the venues nearby that may look interesting to the user. Furthermore, the POI list may be sorted according to the user's preference. In order to transmit nearby venue information to the user, the location server needs to know the user's current location. However, this gives rise to a concern of people's location privacy breach. With plenty of a user's location points, a LBS provider can easily track a user and this track is sometimes unwanted. For example, when a user goes to see some medical specialist or participates in a protest at a particular place, the LBS provider will infer her health condition or easily know her political tendency. Once a LBS provider acquires such private information, people will loose control over their privacy and the provider can use it in any way he likes. The provider may sell the health information to Ads companies and the user's mailbox may be jammed by many spam mails. LBS providers may also hand individual private information to the government against users' will, which may bring some unwanted trouble to a user. Therefore, a privacy preserving LBS system is necessary to address aforementioned concerns.

In this paper, we are going to propose a system to preserve a user's location privacy while still enable the user to retrieve venues of interest and their information. Initially, we came up with an area based approach. Here, the term area is divided by their geographic relationships, e.g., zip code, etc. All venues within the same area divided into the same group. Each venue is assigned with a unique ID by the LBS server. An area's associated venue IDs and their attributes and stored in an entry of a database in the LBS server. Knowing the index to the entry, the user could retrieve all venue IDs within this area using private information retrieval (PIR) [6], [16] such that the location server does not know which area of venue IDs have been retrieved. After retrieving all desired venue IDs, the user run a local recommendation system to find matching venue IDs. However, to obtain the index, a user has to maintain a local map that outputs the index given a specific location point. Like many off-line GPS application, this map is pre-loaded to the smart phone.

Downloading map in advance is not as convenient as requesting service on demand. We propose a cell site based approach to address this issue. To connect to a cellular network, a smart phone always keeps contact to the nearest cell site and a cell site is usually a fixed point in a geographic area. This inspires us to use cell site's transmission range as a representative of area. All venues' information associated to a cell site is stored in database in the server. We use a hash table to map a cell site ID to its corresponding index. Once a smart phone obtains a cell site ID, it computes a hash function which gives an index in the hash table. The smart phone invokes a PIR protocol to obtain the cell site's database index which is used to retrieve the venue information in another PIR protocol execution. We remark that a cell site ID is assigned by Federal Communications Commission (FCC) [4] and every smart phone obtains a cell site ID when it sends probe message to the cell site. Also, in a urban area, we have more cell sites with smaller transmission range than that in the wild. This fact also benefits our design since there are more POIs in urban areas. In this paper, we focus on the cell site based approach.

The contributions of this paper can be summarized as follows.

- We propose to use area representation to divide venues into groups such that the user can retrieve venue IDs while the location server does not know which area of venue IDs have been retrieved.
- We propose to use cell site transmission range as a representative of area such that the user does not need

to download area IDs in advance.
- We demonstrate the efficiency of our proposed system through simulations.

The rest of our paper is organized as follows. We give the problem definition in Section II. We introduce our system construction in Section III. In Section IV, we demonstrate our simulation result. We introduce the relate work in Section V and conclude our work in Section VI.

II. Problem Definition

In this section, we present our system model, threat model, and design goals.

A. System Model

Our system consists of a LBS server LS and multiple users. The LBS server is maintained by a LBS provider, which has an amount of venue information stored on LS. Hereafter, we will not distinguish between LBS provider and the server and use the two terms interchangeably. Since all intersections are between a user and LS, we use u to denote the user intersecting with LS. Each venue has a unique venue ID vid, assigned by LS, and a vector of associated attributes. The attributes consists of this venue's information, such as the venue type (restaurant, coffee shop, etc.), venue location, etc. All venues IDs and their attributes are stored in LS. Each user stores a copy of her preference profile in her smart phone. We define a cell site ID as cid. When a user u arrives at and wants to explore a specific place, her smart phone connects to the nearest cell site, identifying the cell site ID cid, and sends a index query to LS. The LBS server sends back the index to the venue information associated with the nearest cell site. After that, the user uses this index to retrieve the venue information. The smart phone locally compares each vid's attributes with the user's profile, sorts all vid's according to the user's preference, and displays the sorting result to the user.

B. Threat Model

The LS in this work is considered as honest but curious, which is accordance with many security systems [5], [19]. The LBS server honestly follows the system execution in order to maintain its business reputation. However, it is curious about users' private information, e.g., location information, search history, etc., and does whatever it can to figure out the information. The LBS server may collect the messages transmitted between users and itself, analyze them, and infer users' private information. At the same time, we assume the cellular service carrier, also the owner of cell sites, is trusted and honest. A carrier can always track any of its subscriber's location as long as they stay connected. It is impossible to maintain privacy in the presence of a malicious carrier.

C. Design Goals

We introduce our system design goals as follows.

- Location privacy: The LBS server will not infer the user's location when it receives the user's request, while the user can retrieve the information she requested.

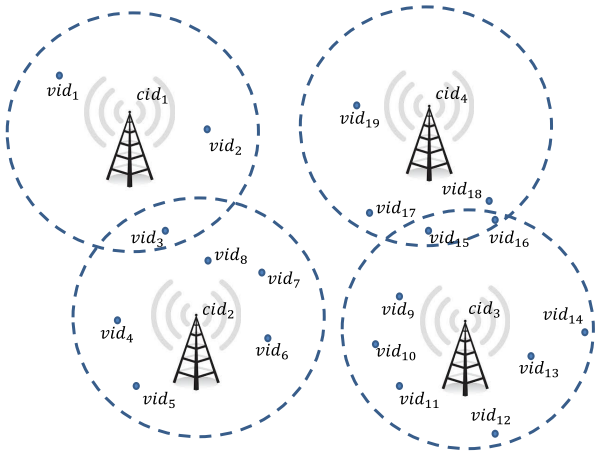

Fig. 1. Using cell site to group venues

- Efficiency: the system should not be computational intensive and each query should be done efficiently.

III. System Construction

In this section, we demonstrate our system construction. First, we introduce the cell site based approach and briefly introduce our system. After that, we demonstrate the PIR protocol. Finally, we give the details of our system construction.

A. Cell site based approach

As we previously introduced, in the area based approach, a user needs to download the area ID in advance in order to retrieve desired venues from the LBS server. In contrast, cell site based approach does not require users to pre-download any help data. The idea stems from the fact that all smart phones have to connect with the nearest cell site at any time to get service. Each cell site has certain transmission range and spaced from 1/4 miles to 2 miles, depending on the density of service subscribers. In the dense urban area, cell sites are spaced from 1/4 miles to 1/2 miles. Therefore, we cluster the venues to the nearest cell site and form an associated venue group. Each cell site ID and its corresponding venue group are stored in the LBS server LS. Connecting a cell site, a smart phone knows the cell site's ID, which can be used in our system to retrieve the venue group.

In Figure 6, we can see that 19 venues are divided into four groups, each of which belongs to one cell site. We note that in the intersection of two areas, there might be venues belonging to more than one area. However, this phenomenon does not affect our system performance. A user may also stay at the intersection of two or more areas. In this case, we regard the area formed by the cell site connected to the user's smart phone as the query area.

As shown in Figure 2, our system has two data structures: a hash table and a venue information table containing all venue information. The index $cind$ is stored in the hash table entry indexed by the hash value of the cell site ID. Each index refers to a row of the venue information table. We call the content

stored in a row of the venue information table as a data block. A data block contains venue IDs and their attributes indexed by cid. We note a venue ID and its attributes might appear in multiple times. If a venue is in more than one area, this venue will belong to more than one group. Given a cell site ID, a user can calculate the hash value and get the index to the entry. By retrieving the entry, the user gets the index to a data block and can retrieve it from LS. Both the two retrievals use PIR.

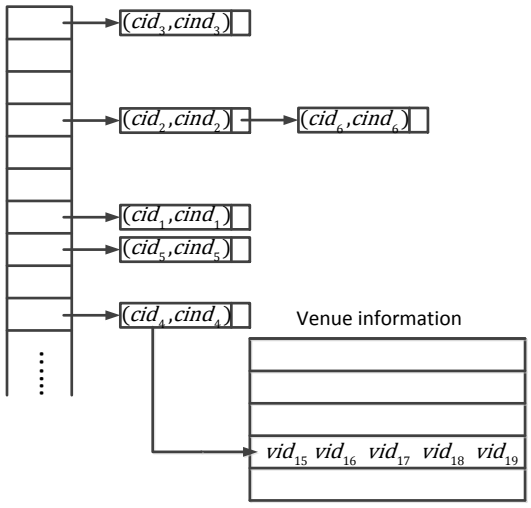

Fig. 2. Illustration of system data structure

B. Initialization

At the beginning of the system running, the LBS server LS needs to do some initialization work. We remark that LS needs to initialize two data sets for retrieval, the hash table and venue information table. In our system, we need to invoke PIR protocol for two times. The user utilizes PIR to retrieve index $cind$ from the hash table stored in the LBS server. Next, using the retrieved index as input, the user uses PIR to retrieve venue IDs and their attributes from the venue information table. Since both retrieval are using the same PIR, we introduce the initialization in a general way. Suppose the total records prepared for retrieval is N. For each record $R_i(1 \leq i \leq N)$ (For convenience, we also use R_i to denote this record's integer representation), the LBS server chooses a unique prime p_i and a appropriate power $v_i = p_i^{c_i}$, such that $R_i < v_i$. We note that p_i is a small prime number. Using Chinese remainder theorem, the LBS server calculates the smallest integer $e \equiv R_i \mod v_i (1 \leq i \leq N)$. The LBS server publishes the integer e and all $\{v_i\}_{(1 \leq i \leq N)}$. Once the initialization is completed, the user can proceed to query the record stored in the LBS server.

C. Using PIR to retrieve a record

Knowing the index v_i to the record R_i, a user runs the following PIR protocol to obtain R_i.

Function: A user retrieves record from the LBS server.
Input: v_i.
Output: R_i.

1. u picks the corresponding index v_i to the record R_i.
2. u generates a group $\mathbb{G}$ and picks a group element g, such that v_i divides the order of g.
3. u computes $q = |\langle g \rangle| / v_i$ and $h_1 = g^q$, and sends $\mathbb{G}$ and g to LS.
4. LS computes $h_2 = g^e$ and sends h_2 to u.
5. u computes $h_3 = h_2^q$ and the desired data $R_i = \log_{h_1} h_3$, where $\log_{h_1}$ is the discrete logarithm base h_1.

Fig. 3. PIR protocol

Figure 3 shows the PIR protocol. In Figure 3, at the last step, the user u needs to computes a discrete logarithm to get the record R_i. It is well known that the calculating discrete logarithm is hard. However, it is feasible to perform the calculation over the power of a small prime. Trivially, we can do brute force to crack it. Or, we can employ a relatively faster algorithm, PohligHellman algorithm [15]. There are many ways to generate group $\mathbb{G}$. We will discuss one of them in Section IV.

Theorem 1. *Using the above protocol, the user can correctly retrieve the record R_i.*

Proof: At the last step of the protocol, the user u computes $R_i = \log_{h_1} h_3$. The expression $\log_{h_1} h_3 = log_{h_1} h_2^q = log_{h_1} g^{eq} = log_{h_1} g^{(|<g>|e/v_i)} = log_{h_1} g^{|<g>|x}$. $log_{h_1} g^{R_i |<g>|/v_i} = R_i$, where x is an integer such that $e = xv_i + R_i$. Thus, the theorem is proved. ■

D. System construction

Our system consists of two protocols: index retrieving protocol and venue retrieving protocol. In the index retrieving protocol, the user retrieves the data block index from the hash table stored in the location server.

Figure 4 shows the index retrieving protocol. In this protocol, from her smart phone, the user learns the connected cell site ID cid_i. User u computes the hash value $h(cid_i)$, and collaboratively run the PIR protocol with the LBS server. The output might be several pairs of cell site ID and its index to the venue information table due to the collision of hash value. The user discards all other useless cell site ID and index pairs, and picks the pair $(cid_i, cind_i)$ corresponding to cid_i.

Function: A user retrieves the index to a venue group

1. u acquires the cell site ID cid_i.
2. u calculates $h(cid_i)$.
3. u and LS invoke the PIR protocol using $h(cid_i)$ as input.
4. From the output of PIR protocol, u picks $(cid_i, cind_i)$.

Fig. 4. Index retrieving protocol

Figure 5 shows the venue retrieving protocol. This protocol

is pretty trivial. First, the user and the LBS server collaboratively invoke the PIR protocol. The input of the protocol is $cind_i$ which is retrieved in the index retrieving protocol. The output is the data block D_i from the venue information table.

Function: A user retrieves venue information.
1 u and LS invoke the PIR protocol using $cind_i$ as input.
2 u gets the data block D_i.

Fig. 5. Venue retrieving protocol

E. Discussion

In our proposed system, we let the user query the venues within the area covered by her nearest cell site. Since our system is motivated by current POI recommendation, there might be some flaws exist in our proposed system. Users who are near the border of the current cell site coverage could be close to various venues in nearby cell site coverage areas. Comparing to those venues far away from her in the current cell site coverage, venues in nearby cell sites might be supposed to be recommended with higher priority. To solve this problem, we let a user's smart phone record previous cell site IDs the user has passed by. The user not only queries the index corresponding to the current cell site ID, but also queries indices corresponding to previous cell site IDs. Using these indices as the input to the PIR protocol, the user can retrieve as many venue as possible near the user. The number of recorded cell site ID is defined by the user. The more cell site ID the user records, the more venues near the user might be retrieved. However, the system takes longer to retrieve the venue information, and the local recommendation system also takes longer time to sort venues. We should make a trade off between the number of recorded cell site IDs and the system running time.

IV. Simulation

In this section, we evaluated the performance of the proposed approach via simulation. We implemented our approach in C++. Crypto++5.6.2 cryptographic library is employed to provide basic cryptographic operations, including group and big integer operations. The implemented system was running on a Macbook pro with 2.53GHz Intel Dual Core CPU and 4GB memory. While we have two approaches — area based approach and cell cite based approaches, we focused and implemented the latter approach in this section.

Here, we need to remark one thing regarding approach implementation. When implementing the PBR protocol, we need to construct a specific group G and a generator g for a given $\pi = p^c$ [7]. Particularly, the order of the generator $r = |\langle g \rangle|$ is divided by π, i.e., $\pi | r$. p_i's are the primes used by server to as moduli for its data blocks and $p_i \neq 2$. In addition, the generator is a psi-generator, i.e. $gcd(|G|/r, \prod_i p_i) = 1$. To generate the group, we generate two primes, $Q_0 = 2q_0\pi + 1$ and $Q_1 = 2q_1 d + 1$, such that q_0, q_1, and d are primes and not previously selected as server's moduli. The generated group G is the multiplicative group over integer $N = Q_0 Q_1$. For g, we just randomly select an element in G and test whether its order is divided by π. Usually, a group element's order is hard to calculate. However, we know N's Euler's totient $\phi(N) = (Q_0 - 1)(Q_1 - 1) = 2^2 q_0 q_1 d\pi$, which makes order calculation trivial.

Cell site number and venue number per area are two factors that directly affect the performance of our approach. Intuitively, increment of the two numbers will increase our approach's processing time. We are interesting in how much the processing time is about to grow, from three aspects — server's initialization time, user's query time, and server's query processing time. User's query time includes her query generating time and response decoding time

First, we increased cell site number from 150 to 1000 with 50 increment and the result performance of our approach is shown in Figure 6. Obviously, the server initialization time

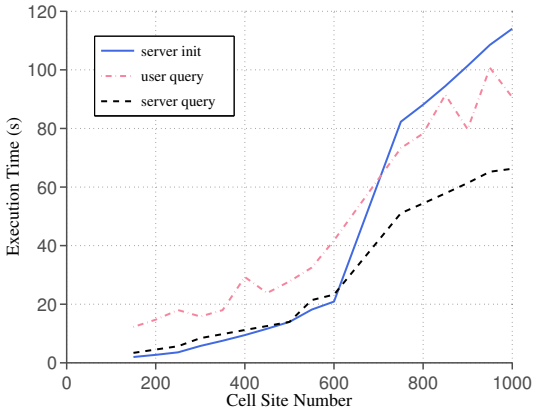

Fig. 6. Execution time vs. cell site number

jumps when there are more than 600 cell sites handled by the approach. It is because the server needs to find a small prime and calculate a suitable exponent for each cell site data block. When cell sites number gets large, this task becomes time consuming because many small primes are consumed so that it is not easy to find a small prime and calculate an exponent for a large prime. Second, user query time and server query time are nearly linear to the increment of cell site number. Besides, the user query time is longer than server query time because the server just does an group element exponentiation in a query processing but a user has to generate a proper group. The group generation involves relative primality test for N and $\prod_i p_i$. This indicates that the overhead on server side is small and a server can handle more user's queries and the same time.

We tested the performance of our approach under different numbers of venue per area. Also, this will increase each entry size of the cell site table in our approach and thus increase the processing time for both user and server. The result is shown in Figure 7. The number of venues grew from 50 to 500 while we kept cell site number as 400. It is clear that all execution time increases linearly to the number of per area venues. We notice that the server initialization time is small

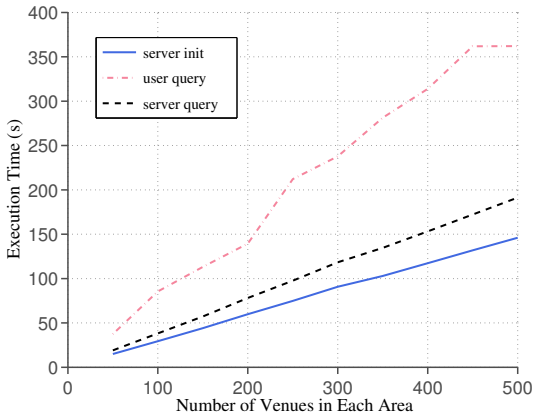

Fig. 7. Execution time vs. number of venues per cell site area

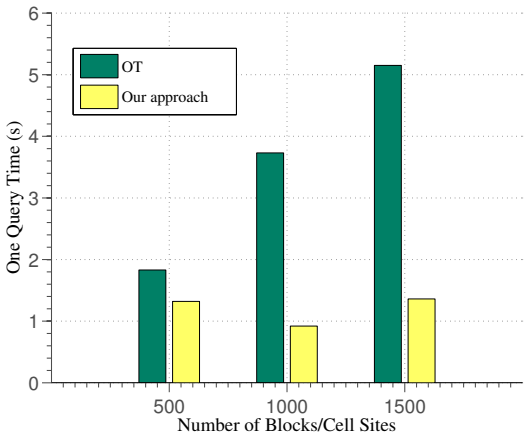

Fig. 8. Comparison between OT based approach and our approach

because the number of cell sites in this test was just 400. Similar to the previous test, the user's processing time is bigger than server processing time. We notice that 300 is a realistic number for per area venues if system is designed in the way that a user is checking venues by categories, such as restaurant, fashion, or entertainment. A supportive fact is that one of one of the biggest fashion malls in Phoenix area has around 250 shops. A user needs 4 minutes to obtain the information of 300 venues. Although 4-minute time looks long, we argue that it is worthwhile to protect people's sensitive privacy. At the same time, the approach can improve the response time by not feeding all the information to the users at once but sending them part by part. When showing the first part, the system does the queries for the remaining part.

We compared our approach with the Oblivious Transfer (OT) based approach proposed in [17]. We note that the difference between our approach and OT based approach lies in the block/cell site index query phase. "Block" is the term used in [17] which is equivalent to cell site area in our approach. Once a user obtains an index, the following step is the same in both approaches — invoking a PBR protocol. The OT based approach used OT to let a user calculate an index while our approach combines hash table and PBR to reduce both computation and communication overhead. We set 300 venues per area/block. The comparison is shown in Figure 8. We can see that when the number of blocks/cell site gets bigger than 1000, our approach is obviously faster than OT based approach. It is because most operations of PBR are cost on one signal data block while that of OT are cost on all data blocks.

V. Related Work

In this section, we review the area of location privacy. Prior studies on location privacy mainly focused on privacy issues in the traditional location based service (LBS), in which a user sends the LBS server a location query and the server returns the local information around the query location. An extensive literature on location privacy utilizes the idea of K-anonymity or location cloaking [2], [9], [14]. In this approach, a user's query is relayed by a trusted third party to the LBS server. The trusted third party does not query the LBS server until K different user queries are collected from the same area. In this way, the LBS server cannot tell which user queries a specific location. The assumption that the third party is fully trusted by users may become a security weakness of this approach. Kalnis *et al.* [11] pointed out that there exists certain scenarios in which K-anonymity approach leaks the private location information to malicious entities. In addition, these approaches employ complex server query processing techniques and entail the transmission of large quantities of intermediate results between users and the trusted third party.

To overcome the disadvantages of cloaking-based techniques, a new class of location privacy protection approaches were proposed, which are known as transformation based approaches. More recently, Yin *et al.* proposed a framework called SpaceTwist [18] to deceive an untrusted location server by incrementally querying nearest neighbor based on their ascending distance from a fake point which is different from the user's actual location. In [12], Khoshgozaran and Shahabi proposed to use a one-way transformation to encode the location query and the object space. The query is evaluated in the transformed space such that the users' location privacy is preserved. Their approaches do not rely on a trusted party, but may have errors in scenarios that require exact result.

Some other works were proposed to preserve the location privacy by using private information retrieval (PIR) [3]. Hengartner proposed an architecture which utilizes PIR and trusted computing to protect user location privacy [10]. However, the proposed architecture is not implemented yet. Ghinita *et al.* used computational PIR to enable private evaluation of the nearest neighbour queries [8]. However, the heavy computational overhead of the underlying PIR scheme makes the approach unsuitable for a smart device. A fast PIR based location privacy scheme was proposed by Khoshgozaran *et al.* [13]. Their scheme requires a tamper-resistant trusted hardware installed close to the server, which makes the scheme less practical. Recently, Albanese *et al.* [1] proposed a novel

approach to detect an attacker's location by using information of detected malicious nodes when the attacker tries to locate a victim in the Mobile Ad Hoc Networks.

Different from the above works, our work is motivated by the POI recommendation, i.e., the user can automatically be recommended with venues of interest when she gets to some specific place. All venues are stored in the location server. The user will not know which venues should be retrieved until the last step. The challenge of our work is to find a way to map venues the user might interested in to something the user could use to communication with the location server. Fortunately, we came up with the cell site based approach, in which the cell site ID is used by the user to retrieve venues of interest from the location server.

VI. Conclusions

In this paper, we have proposed a secure and efficient location based service system by exploring private block retrieval protocol. A user is able to retrieve information of interest associated with the current location without leaking the location to the service provider. We have demonstrated the efficiency of the system via simulations.

References

[1] M. Albanese, A. De Benedictis, S. Jajodia, and P. Shakarian, "A probabilistic framework for localization of attackers in manets," *Computer Security–ESORICS 2012*, pp. 145–162, 2012.

[2] B. Bamba, L. Liu, P. Pesti, and T. Wang, "Supporting anonymous location queries in mobile environments with privacygrid," in *WWW*, 2008, pp. 237–246.

[3] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 965–981, 1998.

[4] FCC, http://www.fcc.gov/, 2014.

[5] J. Freudiger, R. Shokri, and J.-P. Hubaux, "Evaluating the privacy risk of location-based services," in *Financial Cryptography*, 2011, pp. 31–46.

[6] C. Gentry and Z. Ramzan, "Single-database private information retrieval with constant communication rate," in *ICALP*, 2005, pp. 803–815.

[7] ——, "Single-database private information retrieval with constant communication rate," in *Automata, Languages and Programming*. Springer, 2005, pp. 803–815.

[8] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: anonymizers are not necessary," in *SIGMOD Conference*, 2008, pp. 121–132.

[9] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *MobiSys*, 2003.

[10] U. Hengartner, "Hiding location information from location-based services," in *MDM*, 2007, pp. 268–272.

[11] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preserving anonymity in location based services," 2006.

[12] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," in *SSTD*, 2007, pp. 239–257.

[13] A. Khoshgozaran, C. Shahabi, and H. Shirani-Mehr, "Location privacy: going beyond k-anonymity, cloaking and anonymizers," *Knowl. Inf. Syst.*, vol. 26, no. 3, pp. 435–465, 2011.

[14] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: Query processing for location services without compromising privacy," in *VLDB*, 2006, pp. 763–774.

[15] R. A. Mollin, *An Introduction to Cryptography, Second Edition (Discrete Mathematics and Its Applications)*. CHAPMAN & HALL/CRC, 2006.

[16] R. Paulet, M. G. Kaosar, X. Yi, and E. Bertino, "Privacy-preserving and content-protecting location based queries," in *ICDE*, 2012, pp. 44–53.

[17] R. Paulet, M. G. Koasar, X. Yi, and E. Bertino, "Privacy-preserving and content-protecting location based queries," in *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*. IEEE, 2012, pp. 44–53.

[18] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu, "Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services," in *ICDE*, 2008, pp. 366–375.

[19] X. Zhao, L. Li, and G. Xue, "Checking in without worries: Location privacy in location based social networks," in *INFOCOM*, 2013, pp. 3003–3011.