Protecting Privacy in Location-based Services Using *K*-anonymity without Cloaked Region

Zhenqiang Gong
School of Computer Science and Technology
University of Science and Technology of China
Hefei, China
gzqiang@mail.ustc.edu.cn

Guang-Zhong Sun
School of Computer Science and Technology
University of Science and Technology of China
Hefei, China
gzsun@ustc.edu.cn

Xing Xie
Microsoft Research Asia
Beijing, China
xingx@microsoft.com

Abstract—**The emerging location-detection devices together with ubiquitous connectivity have enabled a large variety of location-based services (LBS). Unfortunately, LBS may threaten the users' privacy. *K*-anonymity cloaking the user location to *K*-anonymizing spatial region (*K*-ASR) has been extensively studied to protect privacy in LBS. Traditional *K*-anonymity method needs complex query processing algorithms at the server side. SpaceTwist [8] rectifies the above shortcoming of traditional *K*-anonymity since it only requires incremental nearest neighbor (INN) queries processing techniques at the server side. However, SpaceTwist may fail since it cannot guarantee *K*-anonymity. In this paper, our proposed framework, called KAWCR (*K*-anonymity Without Cloaked Region), rectifies the shortcomings and retains the advantages of the above two techniques. KAWCR only needs the server to process INN queries and can guarantee that the users issuing the query is indistinguishable from at least *K*-1 other users. The extensive experimental results show that the communication cost of KAWCR for *k*NN queries is lower than that of both traditional *K*-anonymity and SpaceTwist.**

Keywords: K-anonymity, privacy, location-based services.

I. INTRODUCTION

Nowadays, location-detection devices—such as cellular phones, GPS-like devices and RFID, etc—are more and more widely used. These location-detection devices together with ubiquitous connectivity have enabled a large variety of location-based services (LBS) which are able to tailor services according to the location of the user requiring the services. Such services mainly rely on *k*-nearest-neighbor queries (*k*NN) [9][12], which retrieve *k* points-of-interest (POIs) closest to the user's location.

Unfortunately, LBS may threaten our privacy. Malicious attacker may collude with LBS provider to steal users' location information and query logs. Assume Bob is the user applying LBS and Alice is the malicious attacker who wants to disclose Bob's privacy. After Alice colludes with the LBS provider to get Bob's location information and his query logs, she can infer Bob's privacy as follows: First, Alice may relate the location information to Bob [1]. To do this, Alice may choose from a variety of techniques such as physical observation of Bob, triangulating his mobile phone's signal, or consulting publicly available databases. If, for instance, Bob uses his phone within his residence, Alice can easily convert the coordinates to a street address (most online maps provide this service) and relate the address to Bob by accessing an online white pages service. Second, Alice can infer Bob's privacy through Bob's query logs.

K-anonymity has been widely studied to protect privacy in LBS. Its main idea is to make the user issuing the query indistinguishable from at least *K*-1 other users. Most existing works [1][2][3][4][5][6][13][14], adopt the framework shown in Fig.1. The user sends its location, query and *K* to the *anonymizer*, which is a trusted third party in centralized systems [1][2][4] or a peer in decentralized systems [5][6][13]. The anonymizer cloaks the exact user location to *K-anonymizing spatial region* (*K*-ASR) including at least *K*-1 other users. Then the anonymizer sends the *K*-ASR and query to the LBS sever, which calculates the candidate results respect to the cloaked region and sends them back to the anonymizer. At last, the anonymizer calculates the actual results and sends them back to the user. Two serious drawbacks of this framework are: 1) high processing cost since the LBS server has to process range *k*-nearest-neighbor queries [1][7], and 2) high communication cost since the number of candidate results can be large.

Different from *K*-anonymity, SpaceTwist [8] sends a false location to the server instead of a cloaked region. SpaceTwist requires only simple query processing algorithm on the server—namely, incremental nearest neighbor (INN) retrieval [9]. However, SpaceTwist may fail if the attacker already knows the locations of all the users. According to [8], the location of the user issuing the query can be bounded in a region Ω. If only one user lies in the region Ω, then attacker can easily infer that the query is issued by the user, which may threaten the user's privacy. The reason why SpaceTwist may fail is that it does not guarantee *K*-anonymity.

In this paper, we presents a framework, called KAWCR (K-anonymity Without Cloaked Region), which aims to improve on the above approaches. Its architecture is shown in Fig.2. The main difference between KAWCR and traditional *K*-anonymity is that KAWCR only sends the center of *K*-ASR to the server while traditional *K*-anonymity sends the whole *K*-ASR to the server. KAWCR only requires server to process INN retrieval, which has been studied extensively [9] and can be readily implemented on the server. In contrast, traditional *K*-anonymity

978-0-7695-4048-1/10 $26.00 © 2010 IEEE
DOI 10.1109/MDM.2010.33

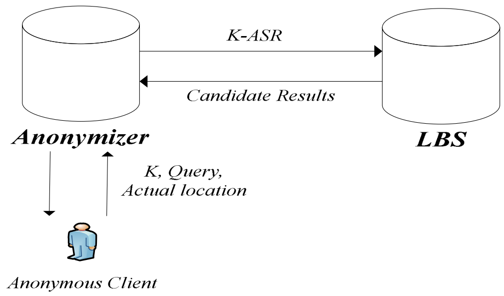


Figure 1. Architecture of traditional *K*-anonymity

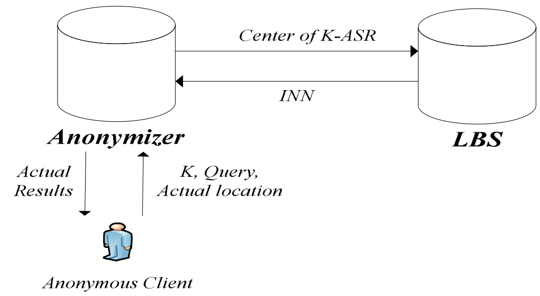


Figure 2. Architecture of KAWCR

requires specialized query processing algorithms, which incur high processing costs. The essential difference between KAWCR and SpaceTwist is that KAWCR can guarantee *K*-anonymity while SpaceTwist cannot.

In this paper, our contributions are as follows:

- We propose a new framework called KAWCR to protect privacy in LBS. KAWCR can guarantee that the user issuing the query is indistinguishable from at least *K*-1 other users with low query processing costs and low communication costs.
- We do extensive experiments to compare KAWCR with traditional *K*-anonymity and SpaceTwist in terms of communication costs. Our experimental results showed that the communication cost of KAWCR is lower than that of traditional *K*-anonymity and SpaceTwist.

The rest of this paper is organized as follows: Section II presents the related works. Next, in Section III, we introduce our approach. Our extensive experimental results are illustrated in Section IV. Finally, in Section V, we conclude this paper and figure out our future works.

II. RELATED WORKS

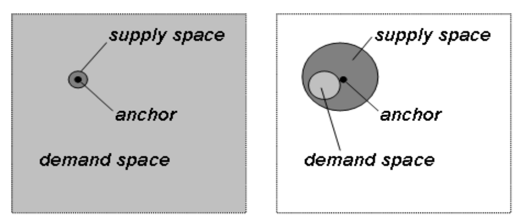
In this section, we review previous works. Although there have existed a variety of approaches to protect privacy in LBS, we focus our mind on traditional *K*-anonymity and SpaceTwist since they are much related to our work. In section II-A, we discuss traditional *K*-anonymity, followed by section II-B, where SpaceTwist is discussed.

A. Traditional K-anonymity

Most existing works on LBSs adopt *K*-anonymity by using the framework illustrated in Fig. 1. This framework works as follows: A user sends its location, query and *K* to the anonymizer, which is a trusted third party [1][2][4] in centralized systems or a peer in decentralized systems [5][6][13]. The anonymizer removes the ID of the user and cloaks the exact user location to *K*-ASR including at least *K*-1 other users. Then anonymizer sends the *K*-ASR and query to the LBS sever, which calculates the candidate results respect to the cloaked region and sends them back to the anonymizer. At last, the anonymizer which knows the locations of all the users calculates the actual results and sends them back to the user. There are two drawbacks of this framework: 1) high processing cost at the server side since the LBS server has to process range *k*-nearest-neighbor queries [1][7], and 2) high communication cost since the number of candidate results can be large.

B. SpaceTwist

Fig.3 provides an overview of SpaceTwist [8], which works as follows: A user specifies an *anchor* and iteratively requests POIs from the server in ascending order respect to the anchor. A circle which is centered at the anchor and includes all the POIs retrieved is called *supply space*. A circle whose center is the user's location and radius is the distance between the user and its current k^{th} nearest neighbor is called *demand space*. In the beginning, the supply space is empty while the demand space is initialized to be the domain space. As POIs are retrieved incrementally from the server, the supply space expands and the demand space shrinks. When the supply space covers the demand space, the algorithm halts and the user is guaranteed to produce accurate *k*NN results. In order to decrease the communication costs, the server accumulates multiple POIs, packs them into the same packet, and then sends the packet to the user.



(a) the beginning (b) the end

Figure 3. Supply Space and Demand Space (cited from [8])

As stated in [8], the location of the user issuing the query can be bounded in an inferred privacy region Ω. If, unfortunately, the malicious attacker knows all the users' locations and only one user lies in the privacy region, then the attacker can infer that it is the user issuing the query, which may lead to the leakage of her/his privacy.

III. K-ANONYMITY WITHOUT CLOAKED REGION

We propose an anonymizer-side *k*NN algorithm that computes *k*NN queries in incremental fashion and guarantees *K*-anonymity. Our proposal only requires the LBS server to support incremental nearest neighbor retrieval, which has been studied extensively [9] and can be readily implemented on servers. In the following, we first describe the anonymizer-side *k*NN algorithm in section III-A. Then, in section III-B, we prove the *K*-anonymity of our proposal, i.e. we prove that the user issuing the query is indistinguishable from at least *K*-1 other users.

A. Anonymizer-side kNN Algorithm

The framework we adopt is illustrated in Fig. 2. A user sends his location, query and *K* to the anonymizer, which calculates *K*-ASR to include at least *K*-1 other users. In order to calculate the *K*-ASR, the anonymizer can use any kind of cloaking algorithm, such as *Interval Cloak* [2], *Casper* [4], *CloakP2P* [6], *NNC* [1], and so forth. Then the anonymizer sends an INN query with the center of *K*-ASR to the LBS server. The server executes INN query processing, which means it iteratively sends POIs back to the anonymizer in ascending distance order respect to the center. After the supply space covers the demand spaces of all the users in the *K*-ASR, the anonymizer halts the INN query on the server. Considering the communication cost, the server accumulates multiple POIs, packs them into the same packet, and sends it back to the anonymizer. Let parameter α denote the number of POIs in a packet. We measure the communication cost of different approaches as the overall number of packets retrieved from the server. In the following, we first discuss anonymizer-side *k*NN algorithm (AS*k*NNA), then prove its correctness.

Anonymizer-side *k*NN Algorithm(AS*k*NNA)

$K^{'} \leftarrow$ the number of users in *K*-ASR
$C \leftarrow$ the center of *K*-ASR, i.e. the anchor
let $q_1, q_2, ..., q_{K^{'}}$ represent the $K^{'}$ users in *K*-ASR
for $i \leftarrow 1$ to $K^{'}$ **do**
  $H_k(q_i) \leftarrow$new max-heap of tuples (q, $dist(p,q)$)
  initialize $H_k(q_i)$ with *k* tuples of ($NULL, \infty$)
  $\beta_i \leftarrow \infty$ ⊳ β_i denotes the radius of the demand space of q_i
end for
$\tau \leftarrow 0$ ⊳ current radius of the supply space
send INN query with C to the LBS server
for $i \leftarrow 1$ to $K^{'}$ **do**
  while $\beta_i + dist(C, q_i) > \tau$ **do**
    $T \leftarrow$ receive the next packet of POIs from server
    $\tau \leftarrow \max_{p \in T} dist(C, p)$
    for all $p \in T$ **do**
      for $j \leftarrow i$ to $K^{'}$ **do**
        if $dist(p, q_j) < \beta_j$ **then**
          update $H_k(q_j)$ and β_j using p
        end if
      end for
    end for
  end while
end for
stop the INN query on the server
return $H_k(q_1)$ to q_1 ⊳ assume q_1 issues the query

Figure 4. Anonymizer-side *k*NN algorithm

The anonymizer-side *k*NN algorithm (AS*k*NNA) is shown in Fig.4 (the same terms have the same meanings as in SpaceTwist). A max-heap $H_k(q_i)$, initialized with k virtual objects, maintains the k nearest POIs of q_i seen so far. Let $dist(p,q)$ denote the distance between point p and q, β_i denote the current radius of the demand space of user q_i, $K^{'}$ denote the actual number of users in the *K*-ASR ($K^{'}$ may be larger than K for some cloaking algorithms[1][2][4]), and τ denote the current radius of the supply space. For each $i = 1,2,...,K^{'}$, if $\beta_i + dist(C, q_i) > \tau$, then the anonymizer continues requesting packets from the LBS server and update $H_k(q_i)$, $H_k(q_{i+1})$, …, $H_k(q_{K^{'}})$ and $\beta_i, \beta_{i+1}, ..., \beta_{K^{'}}$ until $\beta_i + dist(C, q_i) \le \tau$, which means the supply space covers the demand spaces of all the users. At last, the anonymizer terminates the INN query on the server and sends POIs in $H_k(q_1)$ to the q_1 (assume q_1 issues the query).

The following theorem shows that AS*k*NNA finds exact *k*NN results for all the users in *K*-ASR.

Theorem 1 *ASkNNA finds the exact kNN results for each user in K-ASR.*

Proof: For each $i = 1,2,...,K^{'}$, we need to prove that for any POI *p* outside $H_k(q_i)$, inequity $dist(p, q_j) \ge \beta_j$ is satisfied, where β_j is the maximum distance in $H_k(q_i)$. There are two cases depending on whether *p* is seen by the anonymizer or not.

Case 1: p is seen by the anonymizer. Since $H_k(q_i)$ maintains the current *k* nearest neighbors of q_i, we can easily have $dist(p, q_j) \ge \beta_j$, as desired.

Case 2: p is not seen by the anonymizer. In this case, we have $dist(p, C) \ge \tau$, where C is the center of *K*-ASR and τ is the furthest distance respect to C seen so far. Since $\beta_i + dist(C, q_i) \le \tau$ when AS*k*NNA halts, we have $dist(p, C) \ge \tau \ge \beta_i + dist(C, q_i)$, that is, we have $dist(p, C) - dist(C, q_i) \ge \beta_i$. Combined with $dist(p, q_i) \ge dist(p, C) - dist(C, q_i)$, we have $dist(p, q_i) \ge \beta_i$, as desired. □

B. K-anonymity of KAWCR

This section analyzes the privacy protection provided by the KAWCR. We assume that the malicious attacker knows: 1) all the users' locations, 2) the center C, parameter *k*, and all the packets retrieved from the server, and 3) the algorithms used in the anonymizer side. The first assumption is motivated by the fact that users may often issue queries from the same locations e.g. at home or in the office, which may be easily identified through telephone directories, public databases, and so on. Furthermore, users may reveal their locations by issuing queries without privacy requirements. For more discussion about the first assumption, take a look at [1]. The second assumption actually states that either the LBS server or the communication channel between the LBS server and the anonymizer is not trusted. The third assumption is common in the security literature.

Under the above assumptions, the location of the user issuing the query can be bounded in a region. However, we will prove that there are at least *K*-1 other users within this region, which guarantees the *K*-anonymity of KAWCR. Let m be the number of packets received by the anonymizer and let the POIs received (in their retrieval order) be $p_1, p_2, ..., p_{m\alpha}$. Let q_c denote a possible user's location which can be inferred by the attacker, and φ denote the region consisting of all the possible locations q_c. According to the termination condition of AS*k*NNA, the possible user location q_c satisfies:

$$dist(C, q_c) + \min_{1 \le i \le m\alpha}^{k} dist(p_i, q_c) \le dist(C, p_{m\alpha}) \qquad (1)$$

Where the middle term represents the k^{th} smallest distance of all POIs retrieved respect to q_c. The region φ consists of all possible locations q_c satisfying the inequality (1). According to AS*k*NNA, if we replace q_c with any user q_i within *K*-ASR, inequality (1) is also satisfied, which means that all the users within *K*-ASR are also within φ. So the user issuing the query is indistinguishable from at least *K*-1 other users.

IV. Experiments

This section shows our extensive experimental results. Our algorithms are implemented in C++. We perform our algorithms on Dual Core 2.13GHz PC with 2GB memory. In section IV-A, we show the descriptions of the experimental setup. Section IV-B shows the experimental results that compare KAWCR and traditional *K*-anonymity (denoted as TKA) and SpaceTwist in terms of communication costs. We summarize our experimental results in section IV-C.

A. Experimental setup

In our experiments, we use both synthetic datasets and one real dataset. The synthetic datasets are UI and NI. UI means that the coordinates of the POIs are uniformly and independently distributed. NI means that the coordinates of the POIs are normally and independently distributed. The real dataset we use is GN. GN contains 398,958 POIs, which are extracted from U.S. Board on geographic names (*http://geonames.usgs.gov/index.html*). We assume that there are 200,000 users and they are uniformly distributed. In our experiments, the coordinates of POIs and users are normalized to the square with extent 10,000 meters.

TABLE I. Default Values of Parameters Used

Parameters	Default Values
K	100
K	10
d	200m
α	67
Number of users, i.e. *M*	200,000
Number of POIs , i.e. *N*	1,000,000

TABLE I shows the default values of the parameters used in our experiments. In most previous papers, the default value of *k* is set to be 1, such as [1][8]. However, this setting is inappropriate since users may also concern other aspects of the POIs apart from the distances. For example, if a user issues a query to retrieve the interested hotels, he/she may also concern the prices and services of the hotels apart from the distances. So we should return several candidate hotels for the user to choose. In our experiments, the default value of *k* is set to be 10. We set the packet capacity α as (576-40)/8=67. As explained in [8], the typical size of a Maximum Transmission Unit (MTU) over a network is 576 bytes, a 2D data point takes 8 bytes, and a packet has a 40-byte header.

In each experiment, we use workload with 50 uniformly random generated queries and measure the average communication cost, which is the number of packets retrieved from the LBS server.

B. Experimental results

We compare KAWCR with TKA and SpaceTwist in terms of communication costs. In the following, we first show the experimental results of the comparison between KAWCR and TKA, then the results of the comparison between KAWCR and SpaceTwist.

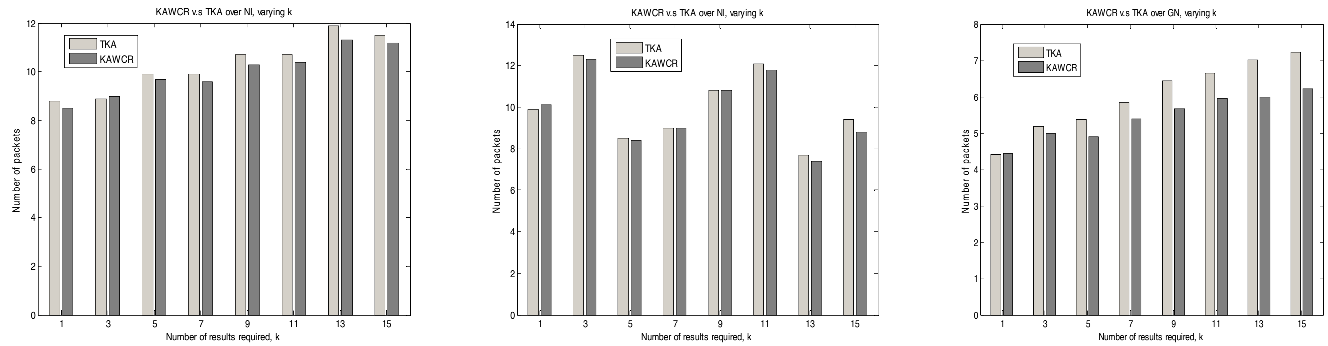
Comparing KAWCR with TKA In our experiments, we use *NNC* algorithm [1] to cloak the user location to *K*-ASR both in KAWCR and TKA. *NNC* algorithm cloaks the user location to a circle including at least *K*-1 users.

Fig.5 shows the experimental results when *k*, the number of results required, varies from 1 to 15 with step 2 and other parameters are set as shown in TABLE I. From the results, we know that the communication costs of KAWCR are lower than those of TKA in most cases. For UI and GN datasets, the communication costs of both KAWCR and TKA slightly increase with *k*. However, it is different for NI dataset. The communication costs of both KAWCR and TKA fluctuate with *k* over NI dataset. As *k* increases, the advantage of KAWCR over TKA increases. The predominance of KAWCR over TKA is more apparent over GN than over UI and NI. For UI dataset, although KAWCR outperforms TKA, the difference between their communication costs is small.

Fig.6 shows the experimental results when *K*, the number of users in *K*-ASR varies from 10 to 100 with step 10 and other parameters are set as shown in TABLE I. From the results, we know that KAWCR outperforms TKA over all the datasets considered. And the predominance of KAWCR over TKA is more apparent over GN dataset, which is a real dataset. The reason why KAWCR outperforms TKA is that TKA retrieves the *k*NN results of all the points in *K*-ASR while KAWCR retrieves the *k*NN results of all the users in *K*-ASR in incremental fashion.

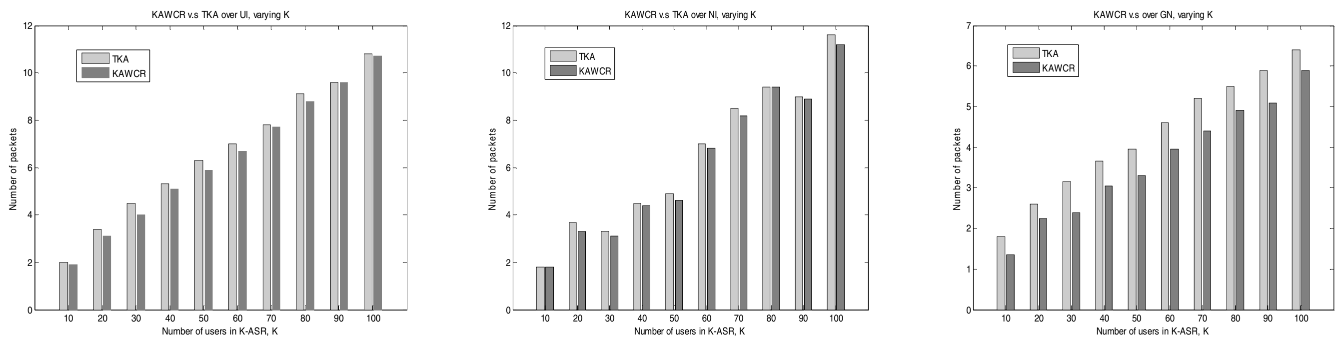
Comparing KAWCR with SpaceTwist According to [8], the location of the user issuing the query in SpaceTwist can be bounded in a region Ω. In order to compare SpaceTwist with our approach fairly, we make the number of users in Ω equate the number of users in *K*-ASR, which means SpaceTwist provides the same level of privacy as KAWCR does under the assumption that the attacker knows the locations of all the users. And we use *NNC* algorithm [1] to cloak the user location to *K*-ASR.

Fig.7 shows the experimental results when *k*, the number of results required, varies from 1 to 15 with step 2 and other parameters are set as shown in TABLE I. From the results, we know that the communication costs of KAWCR are significantly lower than that of SpaceTwist in all cases



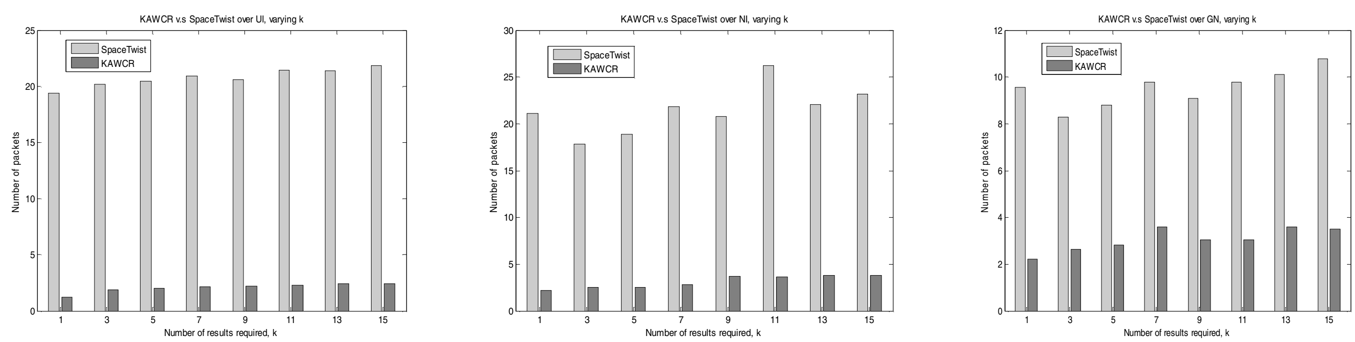
(a) over UI (b) over NI (c) over GN

Figure 5. KAWCR v.s TKA, *K*=100 and *k* varies from 1 to 15



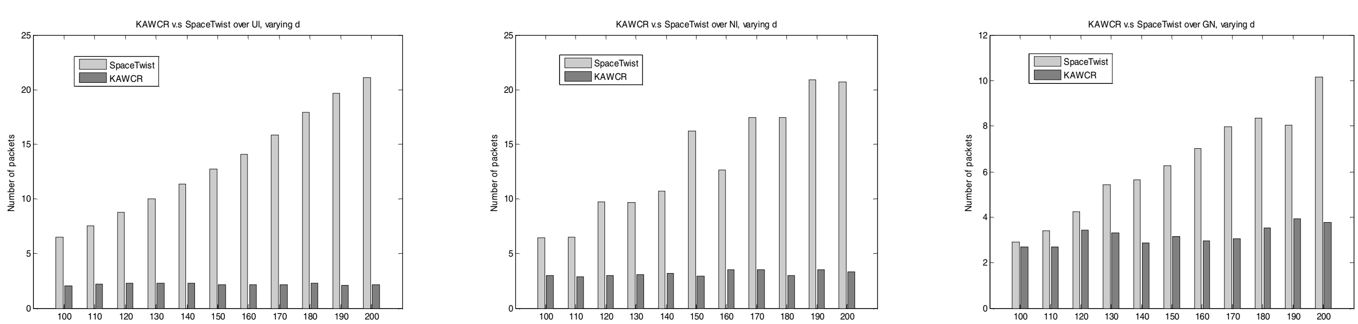
(a) over UI (b) over NI (c) over GN

Figure 6. KAWCR v.s TKA, *k*=10 and *K* varies from 10 to 100



(a) over UI (b) over NI (c) over GN

Figure 7. KAWCR v.s SpaceTwist, *d*=200 and *k* varies from 1 to 15



(a) over UI (b) over NI (c) over GN

Figure 8. KAWCR v.s SpaceTwist, *k*=10 and *d* varies from 100 to 200

considered. The communication costs of KAWCR slightly increase with k over UI and NI while slightly fluctuate with k over GN. And the communication costs of SpaceTwist slightly increase with k over UI while fluctuates with k over NI and GN.

Fig.8 shows the experimental results when d, the distance between the anchor and the user issuing the query, varies from 100 to 200 with step 10 and other parameters are set as shown in TABLE I. The results show that the communication costs of KAWCR are significantly lower than those of SpaceTwist in all cases considered. The communication costs of KAWCR almost keep the same as d increases over UI and NI while slightly fluctuating with d over GN. And the communication costs of SpaceTwist almost increase with d over all datasets considered. As d becomes larger and larger, KAWCR outperforms SpaceTwist more and more. The reason why KAWCR performs much better than SpaceTwist is that SpaceTwist may retrieve the kNN results of other users which are not in the privacy region Ω while KAWCR only retrieves the kNN results of the users in K-ASR.

C. *Summary*

Our experiments illustrate that the communication costs of our proposal KAWCR are lower than those of TKA both on synthetic and real datasets. The reason why KAWCR outperforms TKA is that TKA retrieves the kNN results of all the points in K-ASR while KAWCR retrieves the kNN results of all the users in K-ASR in incremental fashion. From the experimental results, we also know that the communication costs of KAWCR are significantly lower than those of SpaceTwist when they provide the same level of privacy, both on synthetic and real datasets. The reason why KAWCR performs much better than SpaceTwist is that SpaceTwist may retrieve the kNN results of other users which are not in the privacy region Ω while KAWCR only retrieves the kNN results of the users in K-ASR.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we propose a new framework called KAWCR to protect privacy in location-based services. KAWCR can guarantee that the user issuing the query is indistinguishable from at least K-1 other users. Compared with traditional K-anonymity, KAWCR only needs INN query processing algorithm while traditional K-anonymity needs complex processing algorithm at the server side, and the communication cost of KAWCR is lower than that of traditional K-anonymity on some datasets. Compared with SpaceTwist, both KAWCR and SpaceTwist only need INN query processing algorithm at the server side, but the communication cost of KAWCR is significantly lower than that of SpaceTwist on some datasets when they provide the same level of privacy. TABLE II summarizes the three techniques.

TABLE II. SUMMARY OF THREE TECHNIQUES

	KAWCR	Traditional K-anonymity	SpaceTwist
K-anonymity	Yes	Yes	No
Query processing cost	Low	High	Low
Communication cost	Low	High	High

Our proposal considers snapshot k-nearest-neighbor queries. It is interesting to extend it to support continuous queries [15]. Furthermore, we are interested in extending our proposal to support queries in road networks [16].

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China under the grant (No. 60873210) and Anhui Provincial Natural Science Foundation (No.090412064). This work is also partially supported by Microsoft Research Asia Internet Services Theme Research Program.

REFERENCES

[1] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing Location-Based Identity Inference in Anonymous Spatial Queries." In *IEEE TKDE*, 2007.

[2] M. Gruteser and D. Grunwald, "Anonymous Usage of Location-Based Services through Spatial and Temporal Cloaking," In *MobiSys*, 2003.

[3] B. Gedik and L. Liu, "Location Privacy in Mobile Systems: A Personalized Anonymization Model," In *ICDCS*, 2005.

[4] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: Query Processing for Location Services without Compromising Privacy. In *VLDB*, 2006.

[5] G. Ghinita, P. Kalnis, and S. Skiadopoulos, "Priv´e: Anonymous location-based queries in distributed mobile systems," In *WWW*, 2007.

[6] C.-Y. Chow, M. F. Mokbel, X. Liu, "A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Services," In *ACM GIS,* 2006.

[7] H. Hu and D. L. Lee, "Range Nearest-Neighbor Query," In *TKDE*, 2005.

[8] M. L. Yiu, C. 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.

[9] G. R. Hjaltason and H. Samet, "Distance Browsing in Spatial Databases," In *TODS*, 24(2): 265–318, 1999.

[10] P. Samarati, "Protecting Respondents' Identities in Microdata Release," In *IEEE TKDE*, 2001.

[11] L. Sweeney, "K-Anonymity: A Model for Protecting Privacy," In *Int'l J. on Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 5, pp. 557-570, 2002.

[12] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest Neighbor Queries," In *SIGMOD*, pp. 71–79, 1995.

[13] G. Ghinita, P. Kalnis, and S. Skiadopoulos, "MOBIHIDE: A Mobile Peer-to-Peer System for Anonymous Location-Based Queries," In *SSTD*, 2007.

[14] C. Zhang and Y. Huang, "Cloaking Locations for Anonymous Location Based Services: A Hybrid Approach," In *GeoInformatica*, Vol.3, No.2, pp.159-182, 2009.

[15] C.-Y. Chow and M. F. Mokbel, "Enabling Private Continuous Queries For Revealed User Locations," In SSTD, pp. 258–275, 2007.

[16] T. Wang and L. Liu, "Privacy-aware mobile services over road networks." In *VLDB*, 2009.