DCA: The Advanced Privacy-Enhancing Schemes for Location-Based Services

Jiaxun Hua[1], Yu Liu[1], Yibin Shen[1], Xiuxia Tian[3(✉)], and Cheqing Jin[2]

[1] School of Computer Science and Software Engineering, East China Normal University, Shanghai, China
{vichua,leoliu,ybshen}@stu.ecnu.edu.cn
[2] School of Data Science and Engineering, East China Normal University, Shanghai, China
cqjin@dase.ecnu.edu.cn
[3] Shanghai University of Electric Power, Shanghai, China
xxtian@fudan.edu.cn

Abstract. With the popularity of Location-based Services, LBS providers have been obtaining more data, by analyzing which they may infer users' real locations and patterns of behavior. Unfortunately, most previous schemes using *k-anonymity* can hardly resist such fiercer side information-based privacy attacks. To address existing problems, we design a novel metric to accurately measure the resulted privacy level. Additionally, Dual Cloaking Anonymity (*DCA*) and *enhanced-DCA* (*enDCA*) algorithms, which are based on our metric, are also proposed. The former (*DCA*) constructs a k-anonymity set via carefully selecting *k-1* users according to *various query probabilities* of each area and correlations between users' *query preferences*. Then, *enDCA* further employs *caching* and *location blurring* to enhance the privacy preservation. Evaluations show that our proposals can significantly improve the privacy level.

Keywords: LBS privacy · k-anonymity · Confusion degree

1 Introduction

Location-based services are springing up around us, whereas leakages of users' privacy are inevitable during these services. Even worse, adversaries may analyze intercepted service data, and extract more privacy like hobbies, health and property. Hence, privacy preservation is an indispensable guarantee on LBS.

Among existing privacy preservation approaches, ones based on k-anonymity are widely researched. However, some privacy concern will be aroused if these schemes are adopted directly. For example (in Fig. 1), an area is divided into 4×4 cells, where a target user U_t issues a query "Find the nearest hotel" (his privacy profile $k = 4$). *DLS* algorithm [6] selects four blue cells to construct a cloaking set because their *gross query probabilities* are similar. Although such a set reached the maximum entropy, experienced adversaries can exclude some

Y. Cai et al. (Eds.): APWeb-WAIM 2018, LNCS 10988, pp. 63–71, 2018.
https://doi.org/10.1007/978-3-319-96893-3_6

cells if they have richer side information, such as features of each cell and users in the cells.

According to querying features of different cells and U_t's query content, adversaries may exclude cell b & d from the set. With the help of further analyses of query preferences, if adversaries learn that U_t is a businessman, they can confidently locate U_t. Thus, location privacy of U_t is invaded.

To address those defects, we propose a novel privacy metric which first takes into account the impact of richer side information on privacy. Then, *DCA* and *enDCA* algorithms are designed. They both fulfill our objectives while either one has different advantages. Major contributions are summarized as follows:

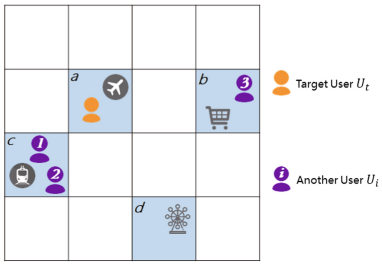


Fig. 1. An example of a cloaking set. More queries about hotels and transport occur in cell a & c, while more queries about entertainment and shopping occur in cell b & d. U_t prefers to query for hotels and conference centers via LBS. U_1 and U_2 mainly search for entertainment.

- A newly-proposed entropy-based privacy metric may measure the privacy level, and depict the impact of richer side information on privacy.
- We design *DCA* algorithm, which considers richer side information (query probabilities & preferences) when constructing k-anonymity sets.
- Based on *DCA*, *location blurring* and *caching* are introduced to *enDCA*. These techniques impede invading location privacy, promote the low bandwidth overhead and resist the disclosure of users' preference privacy.
- We adopt a novel Wi-Fi access point based Peer-to-Peer structure.

2 Related Work

Recently, many research efforts have been concentrated in LBS privacy.

Among cryptography based techniques, Ghinita et al. [2] used Computational PIR, which needs two stages to retrieve POI data. Papadopoulos et al. [10] proposed cPIR which reduces computational overhead.

Kido et al. [3] cloaked user's real location by generating $k-1$ dummy locations, but side information is ignored. *Casper* [5] provided cloaking regions according to user's privacy profile and minimum area, whereas maintaining the pyramid structure leads to high costs. Niu et al. [6,7] designed AP-based k-anonymity schemes considering query probabilities and caching. However, constructing cloaking sets and caching data need high computational and storage overhead for APs, and k-anonymity isn't effectively guaranteed due to negligence in the variety of queries.

Palanisamy et al. [9] constructed adaptive mix-zones centered at road intersections, which replace actual query time with shifted ones, to resist timing attacks. However, these schemes limit the submissions of queries in Mix-zones.

Miguel et al. [1] migrated differential privacy to LBS privacy preservation by adding Laplace noise to users' coordinates.

3 Preliminaries

3.1 Basic Concepts

Query Probabilities. We classify LBS queries into m types with respect to contents of queries. Then we define *various query probabilities* in Eq. 1. For simplicity, an m-dimensional vector $\mathcal{P}_i$ is used to represent respective probabilities of all m types of queries in $cell_i$.

$$\mathcal{P}_i = (p_i^1, p_i^2, \ldots, p_i^m), \quad p_i^j = \frac{\#\text{ of type-}j\text{ queries in } cell_i}{\#\text{ of total queries over all cells}} \tag{1}$$

Users' Query Preferences. Different users have various *query preferences*, which are closely related to their life patterns. We use a vector $\mathcal{W}_i$ to describe the query preference of user U_i (see Eq. 2). Preference vectors will be updated periodically using *Aging Algorithm*.

$$\mathcal{W}_i = (w_i^1, w_i^2, \ldots, w_i^m), \quad w_i^j = \frac{\#\text{ of } U_i's \text{ type-}j\text{ queries (over all cells)}}{\#\text{ of } U_i's \text{ total queries (over all cells)}} \tag{2}$$

Moreover, we use *standardized preference vector* $\mathcal{W}_i^{'} = (w_i^{1'}, w_i^{2'}, \ldots, w_i^{m'})$ instead to preserve users' preference privacy (Different preference vectors may have the same standardized vector), where $w_i^{j'} = \frac{w_i^j - \mu_{\mathcal{W}_i}}{\sigma_{\mathcal{W}_i}}$ ($\mu_{\mathcal{W}_i}, \sigma_{\mathcal{W}_i}$ are the mean and the standard deviation of $\mathcal{W}_i$ respectively). Then, the correlation coefficient between arbitrary two LBS users U_x, U_y is defined in Eq. 3.

$$\rho(U_x, U_y) = \frac{covariance(\mathcal{W}_x, \mathcal{W}_y)}{\sigma_{\mathcal{W}_x} \cdot \sigma_{\mathcal{W}_y}} = covariance(\mathcal{W}_x^{'}, \mathcal{W}_y^{'}) \tag{3}$$

3.2 Adversary Model

In this paper, we resist *eavesdropping attack* performed by passive adversaries via applying SSL on communication channels. We consider LBS servers, who own global data, as active adversaries. Even worse, those untrusted servers may *collude* with malicious users to *infer* normal users' query preferences and behavior patterns by exchanging extra information and analyzing obtained data.

3.3 Privacy Metrics

In order to demonstrate the impact of *query preferences* and *various query probabilities* on privacy quantitatively, we improve the definition of entropy [6].

Supposing a user U_t issues a type-j query in $cell_t$ under the protection of a k-anonymity set. The query preference of U_t is $\mathcal{W}_t$, and the type-j query probability of $cell_t$ is p_t^j. In addition, $k-1$ other users are located in $cell_1, cell_2, \ldots, cell_{k-1}$ (type-j query probabilities of these cells are $p_1^j, p_2^j, \ldots, p_{k-1}^j$). So the *confusion degree* (ξ) of the k-anonymity set is defined in Eq. 4.

$$\xi = -\sum_{i=1}^{k} \rho(U_t, U_i) \cdot q_i^j \cdot \log_2 q_i^j = -\sum_{i=1}^{k} r_i \cdot q_i^j \cdot \log_2 q_i^j \quad (q_i^j = \frac{p_i^j}{\sum_{s=1}^{k} p_s^j}) \tag{4}$$

4 Our Proposed Schemes

4.1 System Model

Figure 2 shows our novel AP-based P2P structure. APs[1] are designed to undertake such *light* workloads as collecting query probabilities, forwarding data, locating users, and storing caches. Maintenance of users' query preference vectors and calculations are conducted by users locally. Besides, LBS users may communicate with APs *anonymously* (i.e. using pseudonyms) to preserve privacy against APs.

4.2 Schemes Overview

We introduce how APs work via the example in Fig. 2. Suppose that Peter issues a query Q in $cell_t$. APs construct an anonymity set by taking following steps.

(1) After an AP receives Q and Peter's real location $cell_t$ (together with $\mathcal{W}'_{Peter}$ and some other parameters), it will determine the query type of Q.
(2) If Q is a type-j query, APs will search for nearby cells with similar type-j query probabilities to $cell_t$. (subject to probability threshold β).
(3) APs forward $\mathcal{W}'_{Peter}$ to users in cells found in step (2).
(4) Any user U_x who has received $\mathcal{W}'_{Peter}$ computes the correlation coefficient $\rho(U_x,\ Peter)$ between his preference vector and Peter's. U_x will reply APs with the coefficient if the value is greater than the preference threshold θ.
(5) APs reply Peter with users who have similar query preferences, together with coefficient values, indexes of probability differences, and indexes of distance between Peter and them. The distance can be measured by # of hops on the grid-based map (e.g. In Fig. 1, the distance between U_t and U_3 is 2).
(6) Peter filters out $k-1$ optimal users locally according to side information above. Then, he will construct a k-anonymity set and issue the formal query.

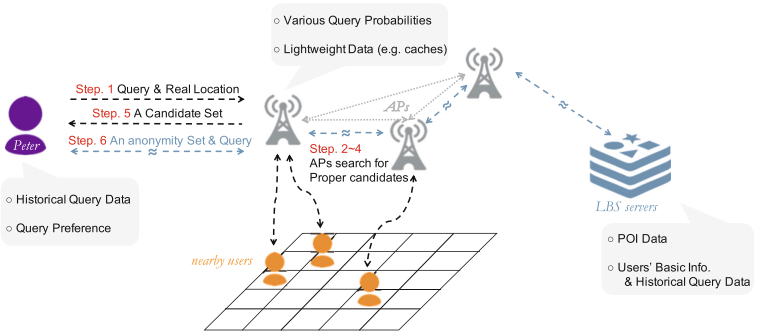


Fig. 2. Schemes overview (data owned by each role is shown in gray blocks)

[1] AP-based schemes [4,6–8] have been widely applied to LBS in mobile environments.

Algorithm 1. Client: DCA Sub-algorithm (issuing a query)

Input: target user U_t's standardized preference vector $\mathcal{W}_t^{'}$, an LBS query $Q(qtype, qdetail)$, real location $cell_t$, privacy profile k_t, distance preference μ, # of sets ns

Output: an optimal k-anonymity set AS

1 send ($\mathcal{W}_t^{'}$, Q, $cell_t$, k_t, μ) to AP (run Algorithm 2);
2 wait until AP returns CS to it; //Alg. 2 (Line 9) shows data structure of CS
3 **for** $(i = 0; i < \min(ns, \binom{3k_t}{k_t - 1}); i++)$ **do**
4 construct set C_i with U_t and $k_t - 1$ other users (in set CS) at random;
5 $score_{C_i} = \sum_{j=1}^{k_t} (index\_prdiff_{ij} \cdot index\_dis_{ij} \cdot r_{ij})$;
6 **return** $\arg\max_{C_i}(score_{C_i})$;

4.3 The Dual Cloaking Anonymity Algorithm

According to the division of work, we implement our schemes in three sub-algorithms. Algorithms 1 and 3 run on clients, and Algorithm 2 runs on APs.

Algorithm 1 demonstrates *DCA* Sub-algorithm which runs on the client of target user U_t (who issues the query actually). It corresponds to Step 1, 6 in last section.

Next, we present Algorithm 2 running on APs. This process corresponds to Step 2, 3, 5 in Sect. 4.2. Index of differences in type-j query probability between the real location $cell_t$ and other cells can be achieved by $index\_prdiff = 1 - \frac{|pr - p_t^{qtype}|}{\beta}$. In addition, we use the index of distance $index\_dis = e^{-\frac{(dis-\mu)^2}{8}}$ to describe users' distance preference. If there aren't enough candidates in CS, AP will extend searching areas (Line 2).

Algorithm 3 computes correlation coefficient between query preferences.

Algorithm 2. AP: DCA Sub-algorithm (forwarding information)

Input: U_t's standardized preference vector $\mathcal{W}_t^{'}$, an LBS query $Q(qtype, qdetail)$, real location $cell_t$, privacy profile k_t, distance preference μ

Output: a candidate set CS

1 CS=NULL;
2 **for** $(d = 1; CS.size() < 3k_t; d++)$ **do**
3 searching for $cell_x$ in d-hop area around $cell_t$, $s.t.\ \forall x,\ |p_t^{qtype} - p_x^{qtype}| < \beta$;
4 send $\mathcal{W}_t^{'}$ to users who are located in these found cells (run Algorithm 3);
5 **while** $\exists$ tuples ($\widetilde{user}$, r) returned from users **do**
6 $index\_dis = e^{-\frac{(d-\mu)^2}{8}}$;
7 pr = getPr($user$, $qtype$); //retrieve the query probability of a cell
8 $index\_prdiff = 1 - \frac{|pr - p_t^{qtype}|}{\beta}$;
9 add tuples ($\widetilde{user}$, r, $index\_dis$, $index\_prdiff$) to CS;
10 **return** CS;

4.4 The Enhanced Dual Cloaking Anonymity Algorithm

We introduce more advanced techniques: *location blurring* and *caching* to *enDCA*, which may upgrade users' privacy at the expense of limited compromise in QoS.

Location Blurring. When applying k-anonymity, the real location is likely to be inferred if k is large, as all dummies are distributed around the real one.

Algorithm 3. Client: compute_corr

Input: U_t's standardized preference vector $\mathcal{W}_t^{'}$, other's preference vector $\mathcal{W}_a$
Output: Pearson correlation coefficient between U_t and himself(herself)

```
1 standardize the vector W_a as W_a';
2 if (r = covariance(W_t', W_a')) > θ then
3 |  return (user~, r);    //user's ID will be replaced by a pseudonym
```

To address that privacy issue, *location blurring* is introduced into *enDCA*. Target user's real location will be shifted to a cell which is randomly selected from the nearby ones (in the 1-hop area) with similar same-type query probabilities.

Caching. Different from previous work [7,11], we propose the idea of caching the anonymity sets. Supposing an LBS user U_a (privacy profile is k_a) issues a query $Q(qtype_a,\ qdetail_a)$. A cached set t can be used to preserve U_a's location privacy if Eq. 5 holds. *Caching* may relieve the workload of APs, reduce the bandwidth overhead, and preserve query preference privacy (reducing transmission of users' preferences). Cache will be maintained by APs in background.

$$\exists t \in AS,\ s.t.\ (1)\ t.qtype = qtype_a;\ (2)\ t.k \geq k_a;\ (3)\ \exists i \in [1,k],\ t.U_i = U_a. \quad (5)$$

The data structure of the cached anonymity sets is as follows:
$AS(qtype,\ k,\ expire,\ U_1,\ U_2,\ \ldots,\ U_k)$, where *expire* is the lifetime of a set.

Algorithm 4. Client: enDCA Sub-algorithm (issuing a query)

Input: U_t's standardized preference vector $\mathcal{W}_t^{'}$, an LBS query $Q(qtype, qdetail)$, real location $cell_t$, privacy profile k_t, distance preference μ, # of sets ns
Output: an optimal k-anonymity set AS (or a cached set CAS)

```
1 send (W_t', Q, cell_t, k_t, μ) to AP (run Algorithm 5);
2 wait until CS or CAS returned from AP ;
3 if CAS != NULL then
4 |  return CAS or a subset of CAS according to k_t;
5 else
6 |  run Lines 3-6 in Algorithm 1 (Client: DCA Sub-Algorithm);
```

Algorithm 4 presents *enDCA* Sub-algorithm which runs on clients. If there exists an appropriate cached set, it'll call Algorithm 1 to construct the set (Line 6).

Algorithm 5. AP: enDCA Sub-algorithm (forwarding information)

Input: U_t's standardized preference vector $\mathcal{W}_t^{'}$, an LBS query $Q(qtype, qdetail)$, real location $cell_t$, privacy profile k_t, distance preference μ
Output: a candidate set CS or a cached anonymity set CAS

1 CS=NULL, T=NULL; //T stores cached anonymity set temporarily
2 **foreach** t in $cache[qtype]$ **do**
3 **if** $t.k \geq k_t$ **and** $(\exists i \in [1, t.k],\ t.U_i == U_t)$ **then**
4 $T = T \bigcup \{t\}$;
5 **if** T != *NULL* **then**
6 **return** $\arg\max_{t \in T}(\frac{\xi_t}{\log_2 t.k})$; //return the set with highest confusion degree
7 run AP: DCA Sub-Algorithm($\mathcal{W}_t^{'}$, Q, shiftLocation($cell_t$), k_t, μ); //run Algorithm 2

Algorithm 5 illustrates *enDCA* Sub-algorithm running on APs. After AP receives U_t's query, it will check in cache whether there exist appropriate anonymity sets. Otherwise, Algorithm 5 shifts U_t's real location first, and then follows ordinary steps to construct a candidate set CS (Line 7).

4.5 Security Analysis (Resistance to Colluding and Inference Attacks)

Adversaries try to infer U_t's real location in the way described in Sect. 3.2. However, the idea of maximizing confusion degree and randomization in our schemes will obstruct their conspiracies. Compared with *DCA*, *caching* in *enDCA* reduces exposure of query preferences. *Location blurring* and *standardized preference vectors* may frustrate their inference of real locations when constructing new anonymity sets.

5 Performance Evaluation

5.1 Simulation Setup

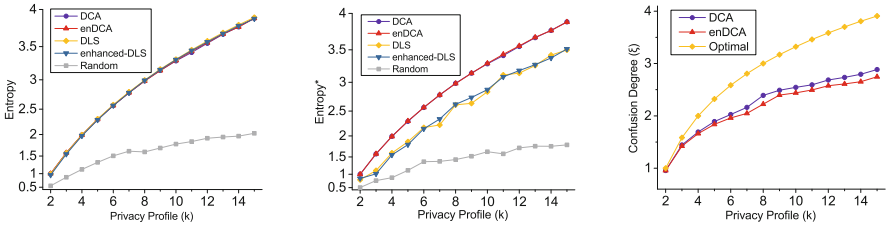
The trajectory data of taxis (From http://soda.datashanghai.gov.cn, involving about 10,000 trajectories) is used to describe the mobility patterns of LBS users in a 10 km × 8 km area in downtown Shanghai. The area is divided into 8,000 cells, with the size of each being 100 m × 100 m. The real deployment of APs in that area will also be simulated. Query probabilities are computed as the users' density in each cell, and the query preferences of users are randomly assigned under normal distribution. Parameters used in our simulation are as follows:

Privacy profile k is set from 2 to 15. # of query types $m = 5$, # of sets $ns = 100$. Threshold $\beta = 0.0015$, $\theta = 0.2$.

We select *Random* [3] as the baseline scheme. *DLS* (*enhanced-DLS*) [6], one of state-of-the-art methods, is also chosen as a comparison.

5.2 Evaluation Results

k vs. Privacy Metrics. Figure 3(a) and (b) show the relation between k and entropy. *Gross query probability* is used in Fig. 3(a), so that all schemes except for *Random* perform well. On the contrary, *various query probability* highlights the advantages of our schemes in Fig. 3(b).



(a) k vs. Entropy (b) k vs. Entropy* (c) k vs. ξ

Fig. 3. Effect of k on privacy metrics

As to confusion degree (Fig. 3(c)), *DCA* edges out *enDCA*, as *enDCA* sacrifices some confusion degree to decrease bandwidth overhead. Our schemes have high but not theoretically optimal results because finding $k - 1$ nearby users having approximately the same query preferences is quite tough.

Other Performance Evaluations. Figure 4 depicts that bandwidth overhead of *enDCA* outperforms *DCA*, since *caching* can serve users' requests for anonymity sets. Figure 5 illustrates the relation among k, cache hit ratio and simulation time t. The hit ratio increases gradually with the t, and smaller k

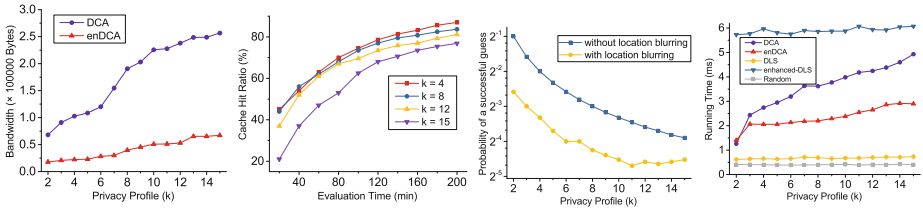


Fig. 4. Bandwidth **Fig. 5.** Cache **Fig. 6.** Guessing Pr. **Fig. 7.** Efficiency

usually results in higher ratio. Figure 6 confirms that schemes without *location blurring* have the theoretical k-anonymity. *enDCA*, equipped with *location blurring*, owns significantly lower probabilities of successful guesses. Figure 7 shows the running time of all schemes. Our schemes consume moderate time to construct a k-anonymity set, and *enDCA* costs less time than *DCA* with the help of *caching*.

6 Conclusion

We propose two different LBS privacy-enhancing schemes, and a novel metric to measure the privacy level. *DCA* constructs a k-anonymity set via carefully selecting $k-1$ users according to various query probability and users' query preferences. Based on that, *caching* and *location blurring* are introduced to *enDCA*, which reduce exposure of query preferences, and decrease the bandwidth overhead. Simulations confirm the effectiveness of our schemes.

Acknowledgment. Our research is supported by the National Key Research and Development Program of China (2016YFB1000905), NSFC (61772327, 61370101, 61532021, U1501252, U1401256 and 61402180), Shanghai Knowledge Service Platform Project (No. ZF1213), Shanghai Science and Technology Committee Grant (15110500700).

References

1. Andrés, M.E., et al.: Geo-indistinguishability: differential privacy for location-based systems. In: 2013 ACM SIGSAC, pp. 901–914 (2013)
2. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.L.: Private queries in location based services: anonymizers are not necessary. In: ACM SIGMOD (2008)
3. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: ICPS, pp. 88–97 (2005)
4. Luo, W., Hengartner, U.: VeriPlace: a privacy-aware location proof architecture. In: ACM SIGSPATIAL GIS, pp. 23–32 (2010)
5. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: query processing for location services without compromising privacy. In: VLDB, pp. 763–774 (2006)
6. Niu, B., Li, Q., Zhu, X., Cao, G.: Achieving k-anonymity in privacy-aware location-based services. In: IEEE INFOCOM, pp. 754–762 (2014)
7. Niu, B., Li, Q., Zhu, X., Cao, G.: Enhancing privacy through caching in location-based services. In: IEEE INFOCOM, pp. 1017–1025 (2015)
8. Okamoto, M., Fujita, N., Inomae, G., Tate, H.: Wi-Fi LBS: information delivery services using Wi-Fi access point location. NTT Tech. Rev. 11(9) (2013)
9. Palanisamy, B., Liu, L.: MobiMix: protecting location privacy with mix-zones over road networks. In: IEEE ICDE, pp. 494–505 (2011)
10. Papadopoulos, S., Bakiras, S., Papadias, D.: pCloud: a distributed system for practical PIR. IEEE TDSC **9**(1), 115–127 (2012)
11. Shokri, R., Theodorakopoulos, G., Papadimitratos, P., Kazemi, E.: Hiding in the mobile crowd: locationprivacy through collaboration. IEEE TDSC **11**(3), 266–279 (2014)