ECM3401: Individual Literature Review and Project

An Investigation into Privacy Preserving Methods of Location Based Services

Charles Roberts

Abstract

Location Based Services (LBSs) have begun to play an important role in almost all aspects of our society. However, the current design is a privacy invasive process as an LBS server is able to obtain all of the information related to a user. This could be used to track users' movements, or even sold to third parties for financial gain. To address this issue, this paper proposes two dummy location generation algorithms to preserve the location privacy of a user during an LBS request. The first algorithm generates k - 1 dummy locations whilst considering the *side information* of the local map. The second enhanced algorithm considers the *privacy area* in addition to the *side information* when the dummy locations are selected. Unlike other approaches, the algorithms proposed in this project require only a privacy degree input from the user. This would allow any user to use either algorithm with no knowledge of the privacy preserving mechanisms being used. Evaluation results show that both algorithms can considerably increase the level of privacy in terms of the *entropy* metric, and the second enhanced algorithm is able to maximise the *privacy area* of the generated location information.

I certify that all material in this dissertation which is not my own work has been identified.

3rd May, 2019

Contents

1 Introduction

With the proliferation of smart phones, tablets and other Internet of Things (IoT) devices, Location Based Services have become increasingly popular and have begun to shape the way we use the Internet today. These services can be used to book a taxi, order food, or even find friends in your area. They are a powerful set of tools that are both simple and straightforward to use. However, the need to disclose your precise location when sending an LBS request raises serious concerns from a privacy perspective. This location information can be used alongside the time and query contents of an LBS request to track users' movements and discover more information about them such as where they live, their lifestyle and even their sexuality. The service providers themselves may even compromise a user's privacy by selling their sensitive information to third parties for advertising purposes. Therefore, it is of vital importance that this user information is not revealed to any untrusted parties when using these services. The use of LBSs themselves is dependent on user's trust. They cannot expand and prosper if people do not trust the services and information they provide.

In order to address this privacy issue, several different approaches have been proposed in previous works. Most of the existing work is focused on location perturbation and obfuscation [1], [2], [3], which employ privacy metrics such as *k-anonymity* [4] to preserve the location privacy of a user. Gruteser *et al.* first introduced the concept of *k-anonymity* in [5], were an adaptive-interval cloaking algorithm is used to 'hide' the user's location within a two-dimensional space that contains at least k_{min} users. However, this scheme relies on a trusted *location anonymizer* [6], [7], which acts as a single point of failure (SPOF) for the approach.

The encountered based solutions proposed in [8] and [3] avoid the *location anonymizer* by transferring information across a Peer-to-Peer (P2P) network of users. The additional resources required for mobile devices to exchange this information however means the scheme would be difficult to implement. To address this problem, Kido *et al.* [9] propose to 'hide' the location of the user within a set of dummy locations. A similar mechanism proposed by Lu *et al.* [10] generates k - 1 dummy locations within a virtual circle or grid that covers the user's location whilst considering the *privacy area* metric. However, these schemes do not consider the *side information* [11], [12], [13], when the dummy locations are selected. This could allow an adversary to eliminate one or more of the submitted locations which would mean *k-anonymity* would no longer achieved.

In this paper two dummy location generation algorithms are proposed: Dummy Location Generation 1 (*DLG* 1), and Dummy Location Generation 2 (*DLG* 2). The first algorithm generates k - 1 dummy locations whilst considering the *side information* of the local map. The second enhanced algorithm aims to also enlarge the *privacy area* of the generated location information.

The remainder of the paper is structured as follows. Section 2 introduces some preliminaries of this paper and Section 3 provides a summary of the Literature Review. Section 4 and 5 detail the design and development of the algorithms respectively and Section 6 describes the testing process that was used. Section 7 explains the experimental design and the performance of the algorithms are then discussed in Section 8 before being assessed in Section 9. The final conclusions of this paper are drawn in Section 10.

2 Preliminaries

This Section defines four basic concepts and three location privacy metrics that will be mentioned throughout this paper before describing the system assumptions and motivations of this project.

2.1 Basic Concepts

Definition (local map): The local map of a user is the two-dimensional area within the user's vicinity that is used during an LBS request.

Definition (*side information*): The *side information* refers to the query probabilities of the different regions within the local map of a user [14].

Definition (query probability): The query probability of a particular location indicates the probability that an LBS request is sent from that location [14]. The sum of the query probabilities within the local map of a user must be equal to 1.

Definition (adversary): An adversary is an entity that is able to obtain sensitive information about a user [13].

The LBS server is considered an adversary because it knows all of the information related to a user, including the query probabilities of the local map and the location privacy preserving mechanism that is being used. The LBS server is then able to use this data to infer more information about the user.

2.2 Location Privacy Metrics

Location privacy metrics are used to quantify the different aspects of location privacy. This enables the performances of different privacy preserving mechanisms to be compared with each other and allows us to draw conclusions from any results that are obtained. Several location privacy metrics have been proposed in previous works.

In [5], Gruteser *et al.* use *k-anonymity* to ensure the genuine location of a user is indistinguishable from at least k - 1 other users, and the degree of anonymity is measured using the size of the anonymity set k. Other variations of this mechanism include *l-diversity* [15], and *t-closeness* [16]. In [17], Jiang *et al.* use anonymity by continually changing the pseudonym of the user and the *entropy* privacy metric [17], [18], [8], [3], [19], [14], is used to measure the ability of an adversary to correlate the different pseudonyms with the same user.

The algorithms proposed in this project use the *k-anonymity* metric to generate k - 1 dummy locations within the vicinity of the user. The *entropy* metric is also used alongside the query probabilities of the local map to decrease the ability of an adversary to identify the user's location from the candidate set k. Finally, the *privacy area* metric is used to increase the size of the area covered by the location information generated by the *DLG* 2 algorithm. A more detailed definition of the *k-anonymity* metric, the *entropy* metric and the *privacy area* metric is given below.

Definition (***k-anonymity***): The value of the *k-anonymity* privacy metric is equal to the number of locations that make up the location information of an LBS request [10].

Definition (***entropy***): The *entropy* of identifying the genuine location from a candidate set of location information k, is defined as

$$H = -\sum_{i=1}^{k} p_i \cdot \log_2 p_i, \tag{1}$$

where p_i is the query probability of the i^{th} location of the candidate set. The aim is to achieve the maximum *entropy* $H_{max} = \log_2 k$, which can only occur when all of the locations within the candidate set have the same query probability $\frac{1}{k}$ [3].

Definition (***privacy area***): The *privacy area* of a set of location information is the convex hull of all the locations within the set. This is the smallest area that contains all of the locations within the set [10].

2.3 System Assumptions

When a user submits an LBS request, their position is determined using either the mobile device or the local network infrastructure. The location information of the user is represented using two-dimensional coordinates (x,y), were x is the latitude and y is the longitude of the user. This project is focused on Location Based Services that do not require the user to provide login details or any other kind of identifying information. The LBS request therefore consists of the location information (x,y), the date and time the request was submitted *datetime*, and the query contents of the request *contents*. The user sends this information to the LBS server via the Internet before receiving the appropriate location based response. The complete process is shown in Figure 1.

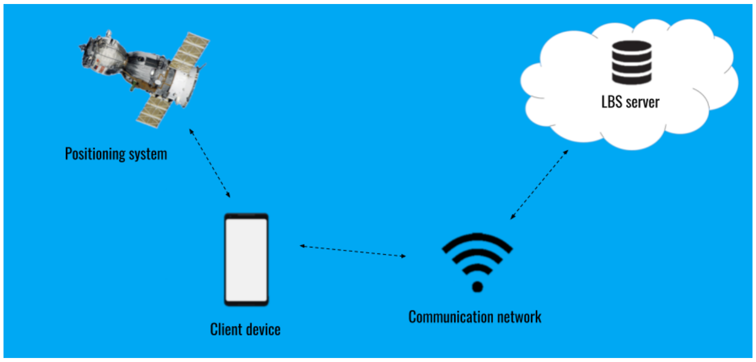

Figure 1: Using GPS to determine the user's location during an LBS request.

2.4 Motivation

One proposed method to preserve the location privacy of an LBS user uses an intermediary *location anonymizer* [20], [21], to increase the area covered by the location information of the LBS request. By reducing the accuracy of the location information in this way, the genuine location is effectively 'hidden' within a two-dimensional region. However, reducing the accuracy of the submitted location information also reduces the accuracy of the LBS response. This is described in more detail in Section 3.2. In addition, the intermediary *location anonymizer* behaves as a performance bottleneck and increases the Trusted Computing Base (TCB) of the system, which in turn decreases the security of the approach.

Other methods use dummy locations to 'hide' the genuine location of the user within a candidate set k of location information [20], [10]. This allows the user to maintain their location privacy whilst achieving an accurate location based response. However, these methods do not consider the *side information* of the local map when the dummy locations are generated. This may lead to some dummies being positioned at locations with a low query probability which could allow an adversary to eliminate certain locations from the candidate set. This would increase the probability of determining the genuine location from $\frac{1}{k}$ to $\frac{1}{k-n}$, where n is the number of eliminated locations. As a result, the *entropy* of the location information would decrease from $\log_2 k$ to $\log_2(k-n)$.

The dummy location methods proposed in [10] and [14] generate k - 1 dummy locations within a virtual circle or grid using a minimum *privacy area* provided by the user along with a value of the *k-anonymity* metric k. In order to use these algorithms however, the user must understand the theory behind the *privacy area* metric and the *k-anonymity* metric for the values of the required inputs to be chosen. Also, using a minimum *privacy area* does not maximise the area covered by the generated location information and so the optimum location privacy is not always achieved.

Two more dummy location generation algorithms were proposed in [13]. The first algorithm generates k - 1 dummy locations using the *entropy* metric and the second enhanced algorithm aims to maximise both the *privacy area* metric and the *entropy* metric when the dummy locations are selected. No minimum area input is required by either of the algorithms proposed in this paper although the functionality of the algorithms must be understood for the value of one of the inputs to be chosen by the user.

The motivation behind the algorithms proposed in this project is to preserve the location privacy of the user by implementing similar mechanisms to those proposed in [13] using only a privacy degree user input *degree*. This would allow the user to achieve the desired degree of location privacy without any knowledge of the location anonymity techniques that are being used. For a location privacy preserving method to be widely adopted it must be as simple to use as the privacy invasive services it protects against.

3 Literature Review

This Section provides a two page summary of the Literature Review.

3.1 LBS Privacy Related Problems

All Location Based Services suffer from the same two privacy related problems: **location privacy** and **query privacy**.

Location privacy: Location privacy refers to the location information of a user and any other private information that can be inferred from it [1].

Query privacy: Query privacy refers to the query contents of the LBS request of a user and any other private information that can be inferred from it [1].

Both location privacy and query privacy are closely related. If the location of a user is obtained, it may be possible to identify them, or, if a user is identified, it may be possible to accurately determine their location. It is possible however for either the location privacy or the query privacy to be compromised while the other is preserved. This is demonstrated in the following two examples:

Example 1: A user is located within a restricted area that contains k - 1 other users of a certain Location Based Service. When the user sends an LBS request, the service provider is unable to identify the users from each other. Therefore, according to the *k-anonymity* privacy metric, the query privacy of the user is preserved. However, because all k users are located within a restricted area, the location of the user can be easily obtained and so the location privacy of the user is compromised.

Example 2: A user is located within an area in which they are the only user of certain Location Based Service. When the user sends an LBS request, the location information covers a wide area which ensures that the service provider is unable to accurately determine the location of the user. Therefore, the location privacy of the user is preserved. However, the query privacy of the user is compromised, as the service provider is able to easily identify the user that sent the LBS request.

3.2 Location Privacy Approaches

According to Matt Duckham *et al.* [22], the strategies used to protect location privacy can be divided up into four different categories: regulatory, privacy policies, anonymity, and obfuscation strategies. This project concentrates on both the anonymity based approaches and the obfuscation based approaches.

Approach	Architecture		*Location anonymizer*	Privacy metric	
	Anonymity based approach	Obfuscation based approach		*k-anonymity*	*Entropy*
Moving in a Neighbourhood [9]	✓	✗	✗	✗	✗
Moving in a Limited Neighbourhood [9]	✓	✗	✗	✗	✗
CirDummy [10]	✓	✗	✗	✓	✗
GridDummy [10]	✓	✗	✗	✓	✗
V-Circle [14]	✓	✗	✗	✓	✓
V-Grid [14]	✓	✗	✗	✓	✓
DLS [13]	✓	✗	✗	✓	✓
enhanced-DLS [13]	✓	✗	✗	✓	✓
Duckham and Kulik: Algorithm 1 [23]	✗	✓	✗	✗	✗
Duckham and Kulik: Algorithm 2 [23]	✗	✓	✗	✗	✗
Gruteser and Grunwald [5]	✓	✓	✓	✓	✗

Table 1: A summary of the location privacy preserving mechanisms examined in the Literature Review.

Obfuscation based approaches: For this approach the quality of the location information of the user is degraded [22]. The idea was first introduced in [23], [24], although similar ideas have also been proposed in other works. In [22], the different obfuscation based approaches were divided up into three mechanisms: inaccuracy, imprecision, and vagueness (see [25], [26], and [27]). A formal definition of imprecision is given below.

Imprecision: The precise location of a user is 'hidden' within a two-dimensional space that covers the location of the user [22]. This is known as a Cloaking Region (CR), and a number of different implementations of this mechanism have been proposed (see [7], [5], and [28]).

The total area covered by the location information sent to an LBS provider is an important factor that must be considered when approaching the problem of LBS privacy. When the user's location is hidden using some Cloaking Region, the total area covered by the submitted location information must be within some range to achieve both location privacy and an accurate location based response.

Anonymity based approaches: For this approach the identity of the user is dissociated from their location information [22]. There are several mechanisms that may be used to achieve this. This paper will focus on one such mechanism which uses dummy locations to 'hide' the user's genuine location from any adversaries (including the LBS provider). This approach combines the user's genuine location with a number of dummy locations generated by a dummy location generation algorithm. All of these locations are then sent to the LBS provider in such a way that the provider is unable to determine which location is genuine. The LBS response corresponding to the user's genuine location can then be filtered out from the responses related to the dummy locations and displayed to the user. This allows the user to achieve an accurate response from the LBS provider whilst maintaining their location privacy.

A summary of all the location privacy preserving mechanisms examined in the Literature Review is given in Table 1.

4 Design

This Section presents Dummy Location Generation 1 and Dummy Location Generation 2 along with Random Dummy Generation, a baseline dummy location generation algorithm that will be used to assess the performances of the two algorithms proposed in this project.

4.1 Dummy Location Generation 1

The objective of the Dummy Location Generation 1 algorithm is to generate $k - 1$ dummy locations within the local map of the user in such a way that they are indistinguishable from the user's genuine location *userpos*. When the user submits an LBS request the local map is divided up into $n \times n$ cells of equal size. Each cell is associated with a query probability p_i, i = 1,2,3,. . . ,n^2, where

$$\sum_{i=1}^{n^2} p_i = 1. \tag{2}$$

The following details the dummy location selection process for the *DLG* 1 algorithm.

(i) The value of the *k-anonymity* metric k is determined using the privacy degree input *degree*. There are three levels of the privacy degree: 'low', 'medium', and 'high'. A greater level of privacy degree is associated with a larger value of the *k-anonymity* metric which provides the user with a higher degree of anonymity and therefore increases the location privacy that is achieved. However, increasing the value of the *k-anonymity* metric also increases the overhead due to the higher computational cost of generating the dummy locations. The user must therefore choose between the speed and low degree of privacy achieved by the 'low' level and the slow but high degree of privacy achieved by the 'high' level. The 'medium' privacy degree is designed to act as a compromise between the other two. The precise values of the *k-anonymity* metric associated with each level of the privacy degree will be decided upon by examining the values of the performance metrics measured during the simulation experiments.

(ii) The query probability of each cell within the local map of the user is computed and used to sort the cells into ascending order. The cell of the user c_{user} is placed in the middle of the cells with the same query probability as the user and the k cells directly above and below the cell of the user are chosen as $2k$ candidate dummy locations. If there are not enough cells below c_{user} then the placement of the cell of the user is changed. The reason that $2k$ candidate dummy locations are chosen is to increase the anonymity degree.

(iii) The algorithm generates k unique sets of cells $C_i = [c_{i1},c_{i2},...,c_{ik}]$, i = 1,2,...,k. Each set is made up of the cell of the user c_{user} and k - 1 other cells that are randomly selected from the $2k$ candidates. The query probabilities of the cells within each set $p_{i1},p_{i2},...,p_{ij},...,p_{ik}$ are then normalised using the formula

$$q_{ij} = \frac{p_{ij}}{\sum_{l=1}^{k} p_{il}}, i = 1, 2, ..., k, \tag{3}$$

to ensure the sum of the query probabilities within each set is equal to 1.

(iv) The *entropy* of each set is calculated using the formula

$$H_i = -\sum_{i=1}^{k} q_{ij} \cdot \log_2 q_{ij}, \tag{4}$$

and the algorithm outputs the set with the greatest *entropy*.

Algorithm 1 shows a more detailed description of the *DLG* 1 algorithm using pseudocode. Despite being able to provide the user with effective *k-anonymity*, the *DLG* 1 algorithm does not consider the *privacy area* of the location information when the dummy locations are generated. This is an important factor that must be taken into account when selecting the dummy locations. It is for this reason that another enhanced version of the algorithm is proposed that tries to spread the dummy locations as far as possible within the local map.

Algorithm 1 Dummy Location Generation 1

```
Input  : $userpos, degree$
Output: optimum set of $k$ cells
determine the value of the $k$-anonymity metric;
determine the values of the query probabilities of the local map;
sort the list of cells according to the query probability values;
position the cell of the user at the appropriate place in the list and choose the $k$ cells right
  before and after the cell of the user to be the $2k$ candidate dummy locations;
$S \leftarrow \emptyset$;
while ($|S| < k$) do
    create a set of $k$ cells $C$ where one cell is the cell of the user and the other $k$ - 1 cells are
      chosen at random from the $2k$ candidate dummy locations;
    $S \leftarrow S \cup \{C\}$;
end
for ($i = 1; i \leq k; i++$) do
    calculate the normalized query probability $p_{ij}$ for each cell $c_{ij}$ in the set $C_i$;
    $H_i \leftarrow -\sum_{j=1}^{k} p_{ij} \cdot \log_2 p_{ij}$;
end
output arg max $H_i$;
```

4.2 Dummy Location Generation 2

The Dummy Location Generation 2 algorithm aims to maximize both the *entropy* and the size of the *privacy area* when the dummy locations are selected. Because two factors are considered, the dummy location selection problem can be formulated as a Multi-Objective Optimization Problem (MOP). This particular MOP is comprised of two objective functions that are optimised simultaneously.

The objective function for the *entropy* metric was defined in Section 2.2. For the size of the *privacy area*, there are two simple approximations that could be used. One is the sum of the distances between consecutive pairs of locations

$$\sum_{i \neq j} d(c_i, c_j), \tag{5}$$

where $d(c_i, c_j)$ represents the euclidean distance between the cells c_i and c_j. The other is the product of the distances between consecutive pairs of locations

$$\prod_{i \neq j} d(c_i, c_j). \tag{6}$$

The example in Figure 2 shows two possible triangles ABC and ABD that could be used to represent the location information of an LBS request. If a triangle is chosen based on the sum of the distances between pairs of locations then either of the triangles could be chosen because $CA + CB = DA + DB$. However, it is clear that the triangle ABC has a larger *privacy area* than the triangle ABD. Using the product of the distances between pairs of locations gives $CA \cdot CB > DA \cdot DB$ and so (6) is used as the objective function for the *privacy area* of this MOP.

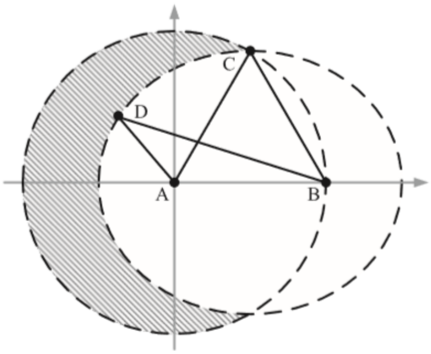

Figure 2: The size of the privacy area [13].

The primary objective of the MOP is to optimise the *entropy* metric

$$D = \arg\max\Bigg(- \sum_{i=1}^{k} p_i \cdot \log_2 p_i \Bigg), \tag{7}$$

and the secondary objective is to optimise the size of the *privacy area*

$$D = \arg\max \prod_{i \neq j} d(c_i, c_j). \tag{8}$$

The MOP can therefore be described as

$$Max\Bigg\{ - \sum_{i=1}^{k} p_i \cdot \log_2 p_i, \prod_{i \neq j} d(c_i, c_j) \Bigg\}. \tag{9}$$

A heuristic solution to this problem is proposed which chooses an optimum set of candidate dummy locations based on the *entropy* metric before selecting k - 1 dummy locations to maximise the size of the *privacy area.* Algorithm 2 shows a more detailed description of the *DLG* 2 algorithm using pseudocode.

The beginning of the algorithm proceeds in a similar manner to *DLG* 1. The n^2 cells are sorted into ascending order according to the query probability values and the $2k$ cells directly above and below the cell of the user are chosen as $4k$ candidate dummy locations. The algorithm then generates k unique sets of cells, each made up of the cell of the user and $2k$ other cells that are randomly selected from the $4k$ candidates. The normalised query probabilities are used to calculate the *entropy* of each set and the randomly selected $2k$ cells within the set with the greatest *entropy* are chosen as $2k$ candidate dummy locations.

The k - 1 dummy locations (c_1,c_2,...,$c_{k\text{-}1}$) are selected in a series of k - 1 rounds. In each round all of the candidates are assigned a certain *weight*, and a candidate is selected with a probability proportional to its *weight.* The probability of a cell c_i with *weight* w_i being selected is therefore equal to $\frac{w_i}{\sum_{j=1}^{x} w_j}$, where x is the number of candidates. In the first round the *weight* of each candidate is its distance from c_{user} and in the proceeding k - 2 rounds the *weight* of a candidate is the product of its distance from c_{user} and the already selected dummy locations.

Algorithm 2 Dummy Location Generation 2

```
Input  : $userpos, degree$
Output: optimum set of $k$ cells
determine the value of the *k-anonymity* metric;
determine the values of the query probabilities of the local map;
sort the list of cells according to the query probability values;
position the cell of the user at the appropriate place in the list and choose the $2k$ cells right
  before and after the cell of the user to be the $4k$ candidate dummy locations;
$S \leftarrow \emptyset$;
while ($|S| < k$) do
    create a set of $2k + 1$ cells $C$ where one cell is the cell of the user and the other $2k$ cells
      are chosen at random from the $4k$ candidate dummy locations;
    $S \leftarrow S \cup \{C\}$;
end
for ($i = 1$; $i \leq k$; $i++$) do
    calculate the normalized query probability $p_{ij}$ for each cell $c_{ij}$ in the set $C_i$;
    $H_i \leftarrow -\sum_{j=1}^{2k+1} p_{ij} \cdot \log_2 p_{ij}$;
end
choose the $2k$ cells within the set with the greatest *entropy* that are not the cell of the user
  to be the $2k$ candidate dummy locations $\hat{D} = \left\{c_1, c_2, ..., c_{2k}\right\}$;
$D \leftarrow c_{real}$;
for ($i = 1$; $i \leq k - 1$; $i++$) do
    choose one candidate dummy location $c_j \in \hat{D}$ to be a dummy location $c$ with a probability
      proportional to $\frac{\prod_{c_l \in D} d(c_j, c_l)}{\sum_{c_j \in \hat{D}} \prod_{c_l \in D} d(c_j, c_l)}$;
    $D \leftarrow D \cup \{c\}$;
    remove $c$ from $\hat{D}$;
end
output $D$;
```

4.3 Random Dummy Generation

The Random Dummy Generation (*RDG*) algorithm randomly generates $k - 1$ dummy locations within the local map of the user at positions which have a query probability greater than zero. This is a baseline dummy location generation algorithm that will be used to assess the performance of the *DLG* 1 and *DLG* 2 algorithms with respect to the *entropy* and *privacy area* of the generated location information. The *computational complexity* of the three algorithms will also be compared with each other by measuring the time taken for the algorithms to complete. Algorithm 3 shows a more detailed description of the *RDG* algorithm using pseudocode.

The *RDG* algorithm first determines the query probabilities of the local map before choosing the cells with a query probability greater than zero (not including c_{user}) as the candidate dummy locations. It then randomly selects k - 1 dummy locations from the candidates and outputs the union of the dummies and c$_{user}$.

Algorithm 3 Random Dummy Generation

```
Input  : userpos,k
Output: random set of $k$ cells
determine the values of the query probabilities of the local map;
choose the cells with a query probability $>$ 0 to be the set of candidate dummy locations $\hat{D}$;
$D \leftarrow c_{real}$;
while ($|D| < k$) do
    choose a candidate dummy location $c_j \in \hat{D}$ to be a dummy location $c$;
    if ($c \notin D$) then
        $D \leftarrow D \cup \{c\}$;
        remove $c$ from $\hat{D}$;
    end
end
output $D$;
```

4.4 Implementation Issues

In order for the *DLG* 1 and *DLG* 2 algorithms to effectively achieve *k-anonymity* the *side information* of the local map of the user must be known before the dummy locations are generated. This Section discusses some of the different mechanisms that have been proposed in previous works in order to obtain this *side information* before the method proposed in this project is presented.

Location Based Services could disseminate the query probabilities directly to their users via the same communication mechanism used during the LBS request. Since query probabilities do not change much over time, the interval between disseminations could be long and so the overhead would not be high. In another approach [13], Wi-Fi Access Points (APs) and users are able to share the *side information* with each other. APs collect the query probabilities within their communication range and users can then download this information and share it with the other APs that they connect to.

In this project the population densities of subjects within different regions of the local map was used to determine the query probability values. It is assumed that a region with a higher population density will have a higher query probability than one with a lower density of subjects. It is proposed that the number of subjects within different regions could be obtained using a similar mechanism to the one proposed in [13]. APs could collect the number of devices within their communication range and users could then download this information and share it with other APs. This is a simple yet powerful mechanism that could be implemented and tested in other projects in the future. A more detailed explanation of how this method was used is given in Section 7.

5 Development

This Section describes why the MATLAB programming language was chosen for this project and discusses some of the issues that were faced during the construction of the dummy location generation algorithms.

5.1 The MATLAB Programming Language

The MATLAB programming language was chosen to implement the algorithms and to create the simulation area of the experiments in this project. Unlike other programming languages, MATLAB is built around matrices which makes it quick and simple to perform complex calculations on large amounts of data. The editor also has a collection of useful tools including a debugger which can be used to pause the execution of a program before it has finished running. This makes it easy to find and fix bugs within the code.

5.2 Development Issues

A. Dummy Location Generation 1

Before the $2k$ candidate dummy locations are chosen by the *DLG* 1 algorithm the cells within the local map of the user are sorted according to the query probability values. The cell of the user is then placed in the middle of the cells with the same query probability as the user.

One problem that was encountered during the development of the *DLG* 1 algorithm however was that the cells either side of c_{user} were not sorted according to the query probability values alone. The position of each cell within the local map was another factor that affected the order of the cells within the sorted list. The result of this defect was that the candidate dummy locations were almost always located within a close proximity of the genuine location of the user. This meant the *privacy area* of the generated location information was small and therefore the location privacy achieved by the algorithm was low. In order to solve this problem the code was modified by shuffling the cells before they were sorted to ensure the position of each cell within the list was based on its query probability value only.

Another more subtle problem was that if the query probability of the genuine location was too high, there were not enough cells directly after c_{user} that could be chosen as the k candidates. However, this was easily solved by checking the position of c_{user} within the list and moving it to k places from the end of the list if necessary.

Two similar problems were found and solved in the *DLG* 2 algorithm code also.

B. Dummy Location Generation 2

After the first dummy location has been selected, the *DLG* 2 algorithm selects the k - 2 other dummy locations from the candidates with a probability proportional to their *weight*. After running the MATLAB implementation of the algorithm however, it was clear that the dummies were being selected with a uniform probability equal to $\frac{1}{n}$, where n is the number of candidate dummy locations. This meant that the distance of each candidate from c_{user} and the already selected dummies was not being considered when the dummy locations were being selected, and therefore the *privacy area* was not being maximised. Solving this problem required the use of the inbuilt MATLAB function `cumsum` to calculate the cumulative sum of the probability values based on the predetermined *weights*.

C. Random Dummy Generation

The original objective of the *RDG* algorithm was to use the *k-anonymity* metric to generate k - 1 dummy locations at random positions within the local map. When the dummies were generated however, some of the positions had a query probability equal to zero. This meant the *entropy* of the generated location information could not be determined because $\log_2 0$ is undefined.

It is for this reason that the objective of the *RDG* algorithm was modified to use both the *k-anonymity* metric and the *entropy* metric to randomly generate dummy locations at positions with a query probability greater than zero. This way the *entropy* of the location information could always be determined and compared to the values achieved by the *DLG* 1 algorithm and the *DLG* 2 algorithm.

6 Testing

This Section details the two testing phases that were performed during the development of the dummy location generation algorithms.

6.1 The Unit Test

The Unit Test is a non-functional test designed to validate whether the individual components or units of a program perform as expected. In this project, the White Box software testing technique was used to traverse different paths through the program code by providing each component with a range of inputs. The functionalities of each component were then observed and modified if any defects were found.

6.2 The System Test

The system test a functional test designed to evaluate the compliance of a fully integrated system with its specified requirements. Here, the Black Box testing technique was used to assess how the different components of each algorithm were able to interact with each other. Several MATLAB scripts were written in order to display the values of certain variables and the location information generated by each algorithm was plotted on a two-dimensional graph. An example of the location information generated by the DLG 2 algorithm for $k = 10$ is shown in Figure 3. In this graph the limits of the x and y axis correspond to the range of the latitude and longitude values of the simulation setup respectively. More information about the simulation set up is given in the following Section.

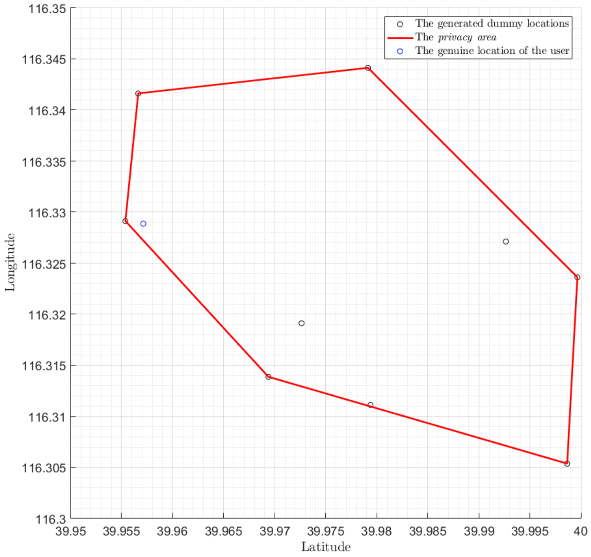

Figure 3: The generated location information of the DLG 2 algorithm for $k = 10$.

7 Experimental Design

This Section describes the simulation setup of the experiments that were performed and the system architecture that was used.

7.1 Simulation Setup

The GeoLife dataset [29], [30], [31], of GPS trajectories was used to simulate the density of subjects within the different regions of the local map. This dataset is made up of the GPS trajectories of 182 users which were logged by Microsoft Research Asia over a period of five years (April 2007 - August 2012).

Most of the trajectories were collected within the City of Beijing which is shown using a heat map in Figure 4, where the brighter areas represent the locations with a higher population density. The simulation area is a square with an area approximately equal to $23.7km^2$ situated within the fifth ring road of Beijing. The latitude coordinates range from 39.95°N to 40.00°N and the longitude coordinates range from 116.30°E to 116.35°E (the white square in Figure 4).

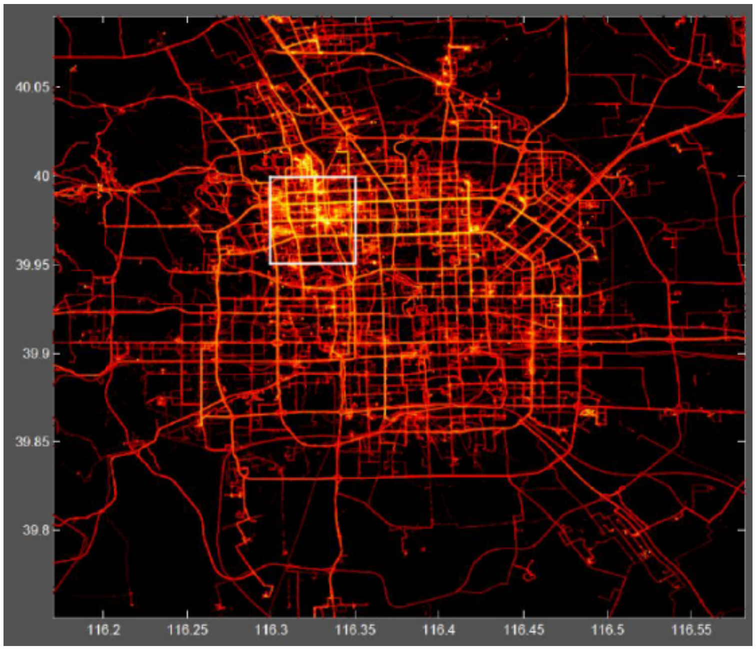

Figure 4: The population density of the City of Beijing [32].

The simulation area was divided up into a 200×200 grid of cells $(c_1,c_2,...,c_{40000})$ represented by the $(latitude,longitude)$ coordinates of the centre of each cell. The latitude, longitude, and time data of the time-stamped points that make up the trajectory information was then used to find the time spent within each cell by all of the users.

7.2 Creating the Simulation Setup

The following details the five stages to creating the simulation setup used in this project.

(i) The position data was read from the multiple '.plt' text files located within the directory of each user in the database using the MATLAB function `textscan`. If the latitude and longitude coordinates of the time-stamp of a user was located within the boundary of the simulation area then the coordinates and time of the time-stamp was stored within an array along with the ID of the user.

(ii) A total of 40000 cells within the local map were created using the MATLAB function `meshgrid` to generate the ($latitude$,$longitude$) coordinates of the centre of each cell. The area of each cell was equal to $\frac{23.688km^2}{40000} = 592.2m^2$, with a side length approximately equal to $24m$.

(iii) The remaining position data was modified to ensure that the time interval between pairs of connective time-stamps for each user was no more than one minute in order to reduce the computational cost of manipulating all of the location information.

(iv) The coordinates of each time-stamp were transformed to the coordinates of the centre of the closest cell. If the set of cells closest to the position of a time-stamp was greater than one then a cell was chosen at random from the set.

(v) The time spent within each cell by each user was found using the coordinates of the time-stamps by calculating the difference between the time values of consecutive time-stamps for each user. The time spent within each cell by all of the users was then found by summing the time spent within each cell by each user.

7.3 Using the Simulation Setup

The dummy location generation algorithms were able to load the simulation data from an array variable stored within a '.mat' file. This was then used to determine the query probabilities of the local map using the formula

$$p_i = \frac{\sum \textit{the time spent within } c_i \textit{ by each user}}{\sum \textit{the time spent within the simulation area by each user}}, \tag{10}$$

where p_i is the query probability of the cell c_i.

For the genuine location input of the algorithms *userpos*, a cell with a query probability greater than zero was chosen at random from the local map.

7.4 System Architecture

The simulation experiments for this project were performed on a HP EliteDesk 800 G4 Tower PC running the 64-bit Windows 10 Enterprise Operating System. The hardware components included 32GB of RAM and an Intel(R) Core(TM) i7-8700 CPU processor with a clock speed of 3.20GHz.

8 Performance Evaluations

This Section discusses the performance metrics used in this project before evaluating the performances of the dummy location generation algorithms and comparing the results with each other.

8.1 Performance Metrics

There were three metrics used to evaluate the performance of the algorithms proposed in this project: the *entropy* metric, the *privacy area* metric and the *computational complexity* metric. A formal definition of the *computational complexity* metric is given below.

Definition (***computational complexity***): The *computational complexity* of an algorithm is a measure of the time taken for the algorithm to complete.

The precise values of the *k-anonymity* metric associated with each level of the privacy degree input for both *DLG* algorithms depends on the measured values of the performance metrics for each value of *k*.

8.2 Evaluation Results

A. Entropy vs. k

The relationship between the *k-anonymity* metric and the *entropy* metric for the *DLG* 1 and *DLG* 2 algorithms is shown in Figure 5. The *entropy* achieved by both algorithms generally increases with the value of *k*. The green line shows the maximum possible *entropy* values $\log_2 k$ that can be achieved for each value of *k*.

The baseline *RDG* algorithm performs the worst as it does not consider the *entropy* metric when generating the dummy locations at positions with a query probability greater than zero. This could allow an adversary to eliminate one or more of the dummy locations from the candidate set and therefore *k-anonymity* would not be achieved.

The *DLG* 1 and *DLG* 2 algorithms achieve much higher *entropy* values than the baseline *RDG* algorithm however because they aim to minimise the range of query probabilities when the dummy locations are generated. The *DLG* 1 algorithm performs particularly well, achieving *entropy* values close to the maximum possible values that can be achieved.

The *DLG* 2 algorithm achieves *entropy* values slightly lower than that of *DLG* 1 because it considers both the *entropy* and the *privacy area* metric when selecting the dummy locations. This means that it must sacrifice the *entropy* in order to maximise the value of the *privacy area.*

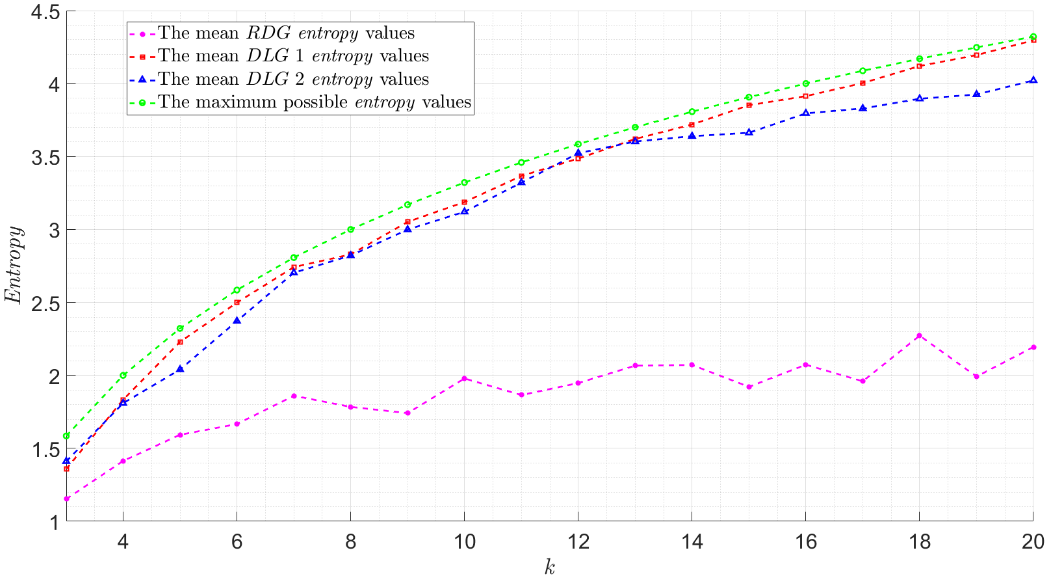

Figure 5: *Entropy vs. k*

B. Privacy area vs. k

Figure 6 shows the relationship between the *k-anonymity* metric and the *privacy area* metric for both the *DLG* 1 algorithm and the *DLG* 2 algorithm. The value of the *privacy area* metric generally increases with *k* for both algorithms as a greater number of dummy locations are more likely to be positioned further apart from each other.

The *RDG* algorithm performs better than the *DLG* 1 algorithm because it does not consider the *entropy* metric when the dummy locations are generated. This means that the *RDG* algorithm is able select the dummy locations from a larger set of candidates and therefore the generated location information is more likely to cover a larger area.

The *DLG* 2 algorithm achieves the greatest *privacy area* for every value of *k* since it considers the distance between the dummy locations before they are selected. The *DLG* 1 algorithm however does not perform as well as the *DLG* 2 algorithm as only the *k-anonymity* metric and the *entropy* metric is used during its execution.

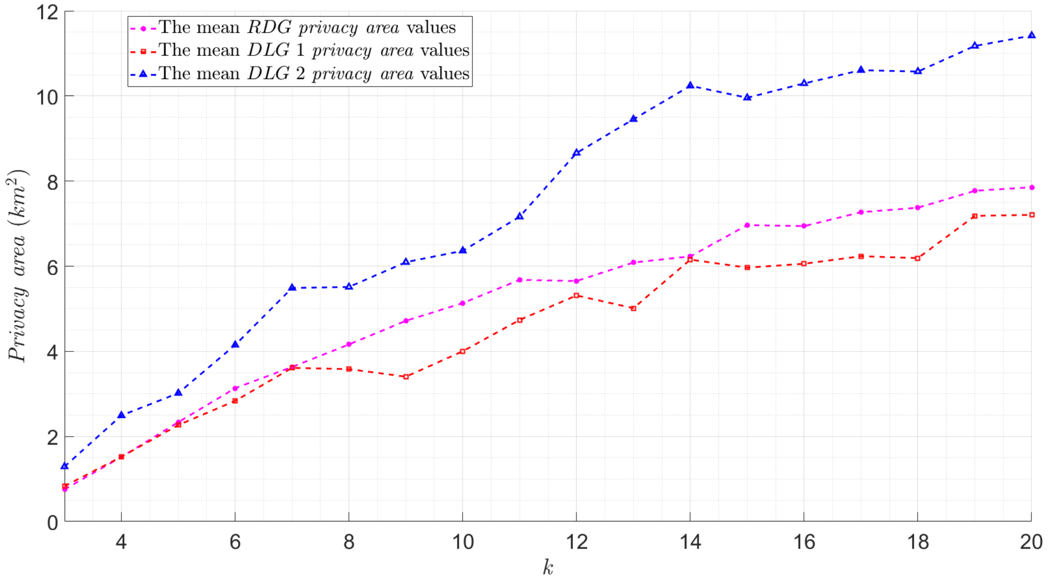

Figure 6: *Privacy area vs. k*

C. Computational complexity vs. k

The relationship between the *k-anonymity* metric and the *computational complexity* metric for the *DLG* 1 and *DLG* 2 algorithms is shown in Figure 7. The baseline *RDG* algorithm consistently achieves the lowest *computational complexity* because it has the lowest computational cost of generating the dummy locations. The running time of the algorithm remains $\approx 0.0049s$ for every value of k.

The greater computational cost of the *DLG* 1 algorithm results in a *computational complexity* slightly higher than the *RDG* algorithm with a ruining time $\approx 0.0156s$ for all values of k.

The *computational complexity* of the *DLG* 2 algorithm however begins approximately equal to the *DLG* 1 algorithm before dramatically increasing to $0.1042s$ when $k = 20$. This is an increase by a factor of 6.5 from $k = 4$ to $k = 20$. The added cost of selecting the dummy locations based on the product of the distances between pairs of locations results in a substantial increase in the running time of the algorithm as the value of the *k-anonymity* metric increases.

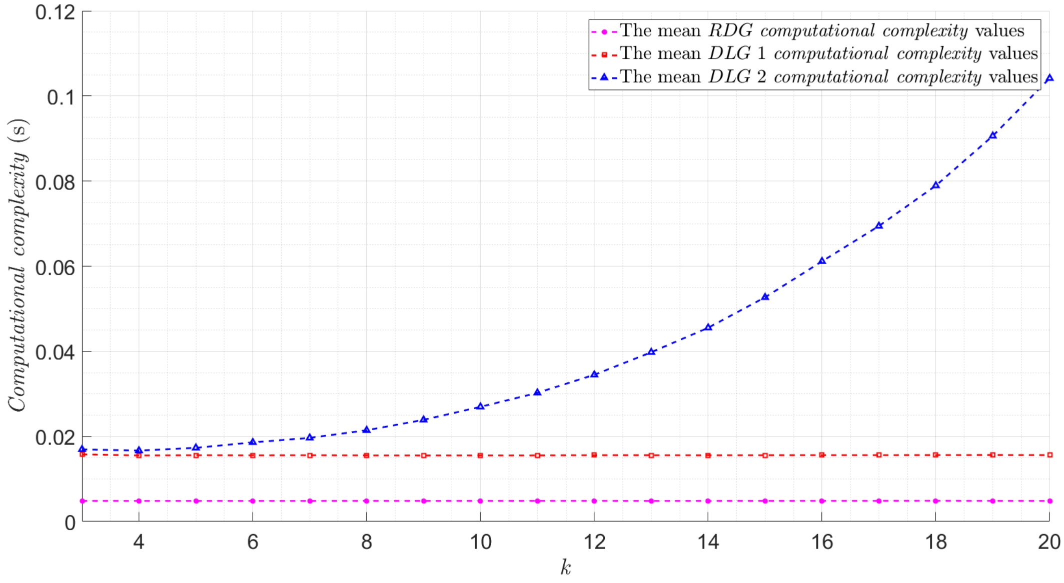

Figure 7: *Computational complexity vs. k*

9 Critical Assessment

This Section critically analyses the algorithms proposed in this project and draws the final conclusions from the experimental results that were obtained.

A. Dummy Location Generation 1

The objective of the *DLG* 1 algorithm is to generate k - 1 dummy locations whilst considering the *entropy* metric using only a privacy degree user input *degree*. A user is able to choose from three increasing degrees of location privacy that are each associated with an increased computational cost. The idea behind this approach is to allow a user to achieve the desired degree of location privacy with no knowledge of the privacy preserving mechanisms that are being used.

The experiments were performed in order to assess the performance of the algorithms proposed in this project and determine the values of the *k-anonymity* metric that would be associated with each level of *degree*. However, the results in Figure 7 clearly show that there is a negligible increase in the *computational complexity* and therefore the computational cost of generating an increased number of dummy locations.

Additionally, from the results shown in Figures 5 and 6 the values of the *entropy* metric remain close to the maximum possible values and the *privacy area* generally increases with *k*. This means that by increasing the number of dummy locations, the *DLG* 1 algorithm achieves a greater degree of location privacy with almost no increase in the *computational complexity*. There would therefore be no disadvantage to the user of increasing the level of the privacy degree input *degree*. Figure 8 shows the same conclusion can be made when the algorithm is run on an iPhone 7 device using the MATLAB mobile application.

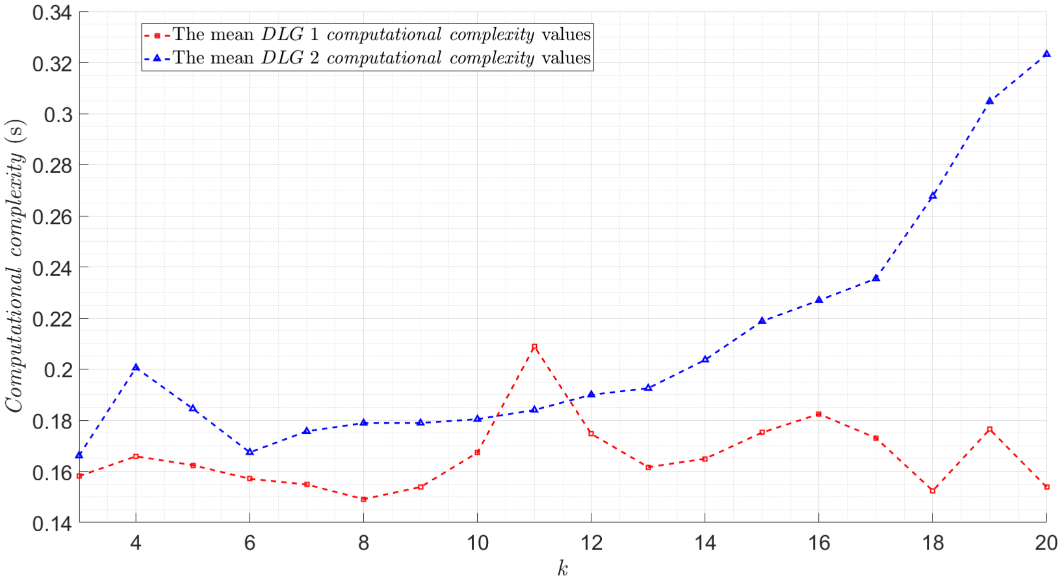

Figure 8: *Computational complexity vs. k* on an iPhone 7.

The usability of the *DLG* 1 algorithm could be further simplified by requiring no input from the user and simply generating the maximum possible number of dummy locations under a certain *computational complexity*. The precise value of this number could be determined by future projects not limited by time constraints using a greater range of the *k-anonymity* values.

B. Dummy Location Generation 2

The objective of the *DLG* 2 algorithm is to generate *k* - 1 dummy locations whilst considering both the *entropy* and the *privacy area* of the generated location information. Like the *DLG* 1 algorithm, the only user input required by the *DLG* 2 algorithm is the privacy degree *degree*, so the user does not need to know the theory behind the location privacy metrics in order to use it.

The *computational complexity* of the *DLG* 2 algorithm dramatically increases with the value of the *k-anonymity* metric as shown in Figure 7. This is because of the additional computational cost of considering the *privacy area* when the dummy locations are selected. Also, Figure 5 shows that the *entropy* values are consistently close to the maximum possible values and Figure 6 shows that the *privacy area* generally increases with the value of k. This means that as the value of the *k-anonymity* metric increases, the location privacy achieved by the *DLG* 2 algorithm also increases along with the *computational complexity*.

However, the *computational complexity* of the *DLG* 2 algorithm is never greater than $0.105s$. This is a negligibly small amount of time for a user to send an LBS request and so there is almost no disadvantage in choosing a greater level for the privacy degree input. The same is true when the algorithm is run on an iPhone 7 device as shown by the blue line in Figure 8 were the *computational complexity* never exceeds $0.34s$.

The usability of the *DLG* 2 algorithm could therefore be simplified in the same way as the *DLG* 1 algorithm by requiring no input from the user. The value of the predefined number of dummy locations would depend on the tolerance of the *computational complexity* and so a greater range of the *k-anonymity* values would have to be tested by projects with more available time.

10 Conclusion

In this paper two dummy location generation algorithms were proposed in order the preserve the location privacy of LBS users. The first algorithm generates k - 1 dummy locations by creating k sets of locations and returning the set with the greatest *entropy*. The second enhanced algorithm uses the same mechanism to choose the optimum set of $2k$ candidates before selecting k - 1 dummy locations whilst considering the *privacy area* of the generated location information. By requiring a privacy degree as the only user input any user is able to use both algorithms without any knowledge of the subject of location privacy.

An AP based solution was proposed to implement these ideas where APs and users are able to share the population density of subjects within different regions with each other. The experimental results show that both algorithms perform well in terms of the *entropy* privacy metric and the *DLG* 2 algorithm can enlarge the *privacy area* whilst achieving a similar privacy level as the *DLG* 1 algorithm. The low *computational complexity* of both algorithms means they could be further simplified although more research is required before any modifications to the functionalities can be made.

References

[1] K. G. Shin, X. Ju, Z. Chen, and X. Hu, "Privacy protection for users of location-based services," *IEEE Wireless Commun.*, vol. 19, pp. 30–39, 02 2012.

[2] S. T. Peddinti, A. Dsouza, and N. Saxena, "Cover locations: Availing location-based services without revealing the location," in *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, WPES '11, (New York, NY, USA), pp. 143–152, ACM, 2011.

[3] B. Niu, X. Zhu, X. Lei, W. Zhang, and H. li, "Eps: Encounter-based privacy-preserving scheme for location-based services," pp. 2139–2144, 12 2013.

[4] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, pp. 557–570, Oct. 2002.

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

[6] C.-Y. Chow, M. F. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*, GIS '06, (New York, NY, USA), pp. 171–178, ACM, 2006.

[7] C.-Y. Chow, M. F. Mokbel, and W. G. Aref, "Casper*: Query processing for location services without compromising privacy," *ACM Trans. Database Syst.*, vol. 34, pp. 24:1–24:48, Dec. 2009.

[8] B. Niu, X. Zhu, H. Chi, and H. Li, "3plus: Privacy-preserving pseudo-location updating system in location-based services," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 4564–4569, April 2013.

[9] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *ICPS '05. Proceedings. International Conference on Pervasive Services, 2005.*, pp. 88–97, July 2005.

[10] H. Lu, C. Jensen, and M. Lung Yiu, "Pad: privacy-area aware, dummy-based location privacy in mobile services," pp. 16–23, 01 2008.

[11] C. Y. T. Ma, D. K. Y. Yau, N. K. Yip, and N. S. V. Rao, "Privacy vulnerability of published anonymous mobility traces," *IEEE/ACM Transactions on Networking*, vol. 21, pp. 720–733, June 2013.

[12] X. Liu, K. Liu, L. Guo, X. Li, and Y. Fang, "A game-theoretic approach for achieving k-anonymity in location based services," in *2013 Proceedings IEEE INFOCOM*, pp. 2985–2993, April 2013.

[13] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, “Achieving k-anonymity in privacy-aware location-based services,” in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 754–762, April 2014.

[14] B. Niu, Z. Zhang, X. Li, and H. Li, “Privacy-area aware dummy generation algorithms for location-based services,” in *2014 IEEE International Conference on Communications (ICC)*, pp. 957–962, June 2014.

[15] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “L-diversity: Privacy beyond k-anonymity,” *ACM Trans. Knowl. Discov. Data*, vol. 1, Mar. 2007.

[16] N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: Privacy beyond k-anonymity and l-diversity,” vol. 2, pp. 106 – 115, 05 2007.

[17] T. Jiang, H. J. Wang, and Y.-C. Hu, “Preserving location privacy in wireless lans,” in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, MobiSys ’07, (New York, NY, USA), pp. 246–257, ACM, 2007.

[18] J. Meyerowitz and R. Roy Choudhury, “Hiding stars with fireworks: Location privacy through camouflage,” in *MobiCom’09 - Proceedings of the Annual International Conference on Mobile Computing and Networking*, Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM, pp. 345–356, 11 2009.

[19] X. Zhu, H. Chi, B. Niu, , , and H. Li, “Mobicache: When k-anonymity meets cache,” in *2013 IEEE Global Communications Conference (GLOBECOM)*, pp. 820–825, Dec 2013.

[20] M. F. Mokbel, C.-Y. Chow, and W. Aref, “The new casper: A privacy-aware location-based database server,” pp. 1499 – 1500, 05 2007.

[21] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, “The new casper: Query processing for location services without compromising privacy,” in *Proceedings of the 32Nd International Conference on Very Large Data Bases*, VLDB ’06, pp. 763–774, VLDB Endowment, 2006.

[22] M. Duckham and L. Kulik, “Location privacy and location-aware computing,” in *Dynamic and Mobile GIS*, pp. 63–80, CRC press, 2006.

[23] M. Duckham and L. Kulik, “A formal model of obfuscation and negotiation for location privacy,” in *Proceedings of the Third International Conference on Pervasive Computing*, PERVASIVE’05, (Berlin, Heidelberg), pp. 152–170, Springer-Verlag, 2005.

[24] M. Duckham and L. Kulik, “Simulation of obfuscation and negotiation for location privacy,” in *Spatial Information Theory* (A. G. Cohn and D. M. Mark, eds.), (Berlin, Heidelberg), pp. 31–48, Springer Berlin Heidelberg, 2005.

[25] M. F. WORBOYS and E. CLEMENTINI, “Integration of imperfect spatial information,” *Journal of Visual Languages Computing*, vol. 12, no. 1, pp. 61 – 80, 2001.

[26] M. Duckham, K. Mason, J. Stell, and M. Worboys, "A formal approach to imperfection in geographic information," *Computers, Environment and Urban Systems*, vol. 25, pp. 89–103, 01 2001.

[27] M. Worboys and M. Duckham, *GIS: A Computing Perspective, 2Nd Edition.* Boca Raton, FL, USA: CRC Press, Inc., 2004.

[28] H. Hu and J. Xu, "Non-exposure location anonymity," in *2009 IEEE 25th International Conference on Data Engineering*, pp. 1120–1131, March 2009.

[29] X. a. and Xie, "Mining interesting locations and travel sequences from gps trajectories," April 2009. WWW 2009.

[30] X. a. and Xie, "Understanding mobility based on gps data," September 2008.

[31] X. a. and Xie, "Geolife: A collaborative social networking service among user, location and trajectory," *IEEE Data(base) Engineering Bulletin*, June 2010.

[32] H. and Fu, X. Xie, , and Q. Li, *Geolife GPS trajectory dataset - User Guide*, geolife gps trajectories 1.1 ed., July 2011. Geolife GPS trajectories 1.1.