LOCATION OBFUSCATION AND DISTANCE-BASED ATTACKS ON PRIVATE TRAJECTORIES: AN EXPERIMENTAL EVALUATION ON REAL TRAJECTORY DATA SETS

by

Aslı Kaya

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of
Master of Science

Sabancı University
January 2015

LOCATION OBFUSCATION AND DISTANCE-BASED ATTACKS ON PRIVATE TRAJECTORIES: AN EXPERIMENTAL EVALUATION ON REAL TRAJECTORY DATA SETS

APPROVED BY:

Assoc. Prof. Dr. Yücel SAYGIN .....
(Dissertation Supervisor)

Assoc. Prof. Dr. Cem GÜNERİ .....

Assoc. Prof. Dr. Hüsnü YENİGÜN .....

DATE OF APPROVAL: .....

Acknowledgements

My sincere gratitude is to my supervisor Yücel Saygın for his guidance, patience, and immense knowledge.

I would like to thank Emre Kaplan for his guidance throughout my thesis studies.

I am thankful to my thesis committee members for their comments.

Last but not least, I would like to thank my family for their endless love, and support.

LOCATION OBFUSCATION AND DISTANCE-BASED ATTACKS ON PRIVATE TRAJECTORIES: AN EXPERIMENTAL EVALUATION ON REAL TRAJECTORY DATA SETS

Aslı Kaya

Computer Science and Engineering, Master's Thesis, 2015

Thesis Supervisor: Yücel SAYGIN

Keywords: privacy, spatio-temporal data, obfuscation, trajectories, data mining

Abstract

With the wide availability of GPS-enabled mobile devices, spatio-temporal data is being collected and stored for providing location-based services or for data analytics. Location-based advertisement, market research, and data mining are just some of the motivations for collecting spatio-temporal data. However, location data is also very sensitive since it may reveal information about the data subject such as his/her political views, religion, state of health, and various personal preferences, which is considered private information. One of the approaches to protect sensitive location data is obfuscation. In this thesis, we have implemented two location obfuscation techniques, performed an analytical and experimental study to investigate how effective they are on a state of the art attack algorithm designed for spatio-temporal data. In the attack scenario, given a set of known trajectories, and a distance matrix composed of known pairwise distances between trajectories, adversary tries to approximate the target trajectory and then extract information about absence or presence of the trajectory in a given area. We used obfuscation techniques to hide information around predefined sensitive places such as hospitals, medical centers. We then used obfuscated data on the attack. Experimental results show that the applied obfuscation methods do not help protecting the privacy of users in sensitive areas in case of spatio-temporal trajectories that follow a regular

pattern. We observed that the attack method works successfully because the obfuscation techniques do not scatter the sensitive points far enough from sensitive places and the linearity of the trajectory is preserved.

KONUM GİZLEME VE ÖZEL YÖRÜNGELER ÜZERİNDE UZAKLIK TEMELLİ SALDIRILAR: GERÇEK YÖRÜNGE VERİ KÜMESİ ÜZERİNDE DENEYSEL DEĞERLENDİRME

Aslı Kaya

Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans Tezi, 2015

Tez danışmanı: Yücel SAYGIN

Anahtar Kelimeler: gizlilik, mekan-zaman verisi, gizleme, yörüngeler, veri madenciliği

Özet

GPS-etkin mobil cihazların geniş ulaşılabilirliği sayesinde mekan-zaman verileri toplanıyor ve konum temelli servislere temin etmek için veya veri analizi için depolanıyor. Konum-temelli reklamcılık, pazar araştırması ve veri madenciliği mekan-zaman verisinin toplanmasında motivasyonlardan sadece bir kaçı. Fakat, konum bilgisi ayrıca çok hassas bir bilgi çünkü özel bilgi olarak düşünülen politik görüş, din, sağlık durumu ve çeşitli kişisel tercihler hakkında bilgileri açığa çıkarabilir. Hassas konum bilgilerini koruma yöntemlerinden biri gizlemedir. Bu tezde, iki konum gizleme tekniğini uyguladık, mekan-zaman verisi için tasarlanan modern bir saldırı algoritmasında ne kadar etkili olduğunu araştırmak amacıyla analitik ve deneysel çalışmalar yürüttük. Saldırı senaryosunda, karşı taraf, bir grup yörüngeler ve yörüngeler arasındaki ikili uzaklıklardan oluşan uzaklık matrisi bilinirken hedeflenen bir yörüngeyi tahmin etmeye çalışıyor ve daha sonra belirli bir alanda yörüngenin varlığı veya yokluğu hakkında bilgi çıkarıyor. Gizleme tekniklerini önceden tanımlı hastaneler, tıp merkezleri gibi hassas yerler etrafındaki bilgiyi saklamak için kullandık. Deney sonuçları gösteriyor ki uygulanan gizleme teknikleri mekan-zaman yörüngelerinin düzenli bir paterni takip etmeleri durumunda hassas bölgelerdeki kullanıcıların mahremiyetini korumak konusunda yardımcı olmuyor. Gizleme teknikleri

hassas yörünge noktalarını hassas yerlerin yeterince uzağına dağıtmadığı için ve yörüngelerin lineerliği korunduğu için saldırı metodunun başarılı bir şekilde çalıştığını gözlemledik.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Privacy issues in spatio-temporal data collected have become more of a concern due to the availability of mobile devices with embedded GPS such as smart phones, phablets and tablets. Spatio-temporal data is collected by mobile service provider companies for various purposes and mostly for data analytics. On the other hand, spatio-temporal data is also very sensitive since it contains information about where a pedestrian/vehicle passed by or spent some time. Places stopped or visited could be a meeting place, medical facility, religious building, which may reveal personal information about the data subject such as his/her political views, religion, state of health or personal preferences. Especially health-status, religion, and sexual life are considered private information and protected by regulations in US, Europe, and many other countries with an established data protection regulation.

Collected spatio-temporal data might be requested by other companies, for market research, advertisement or data mining in general. When a third party requests this sensitive spatio-temporal data, then privacy preserving techniques should be applied before releasing this data. Privacy preserving techniques for spatio-temporal data can be classified into three main groups: (1) access-control policies, (2) anonymity, and (3) obfuscation. Access-control policies are the classical way of protection. Parties that hold permission of the user can have access to personal information. However, if the permission holder does not need whole personal information, it would be an unnecessary disclosure of data. Furthermore, whether giving permission or not, it is not flexible for an entity that provides service according to privacy preferences of user. k-anonymity is another technique, in which a user's data is cloaked, usually through generalization, such that he/she is indistinguishable from

at least k-1 other users. In this way, his/her identity is preserved. This approach needs a third party to be involved and make the necessary changes before releasing the data, which may not be trusted. Besides, this technique does not apply for areas in which it is hard to obtain k users within a reasonable distance in case of location data. Obfuscation techniques are applied via degrading the precision or accuracy ([3]). Although user's identity is revealed, his/her true location is either changed or generalized, thus privacy of the user is preserved. It has a trade-off between the quality of location information, which is important for Location-Based Services (LBS) to provide good service, and privacy of the user.

In this thesis, we concentrate on two obfuscation techniques. One of them is a method that we designed, while the other one is a state of the art technique available in the literature. We implemented those techniques as a tool with a visualization feature to identify points of interests and sensitive locations together with the data subject's trajectory. Later, we attacked the obfuscated data and analyzed the outcome of the attack.

Our method for obfuscation perturbs sensitive data points, provided that the sensitive locations are specified as points of interest. The obfuscation is done such that the user does not appear to be in those locations. Therefore, our method introduces inaccuracy to the data. We have the trajectory, which has points ordered in time; a sensitive point is obfuscated according to its non-sensitive next point such that a circle of candidate data points around this next point is formed. This circle has a radius of average neighboring distance of trajectory points. Then they are mapped to the nearest road segment. One of the two non-sensitive points, having the smallest road distance from the original sensitive point is selected randomly. Thus, the method takes the direction of the movement into account, uses map information and includes randomization to provide further protection against attacks.

The alternative obfuscation technique we considered is the one proposed in [1]. In this technique, location points are treated as circles because of the imprecision of location sensing technologies. Privacy of the location measurement is expressed as a metric, which is independent of the sensing technology, depending on the initial measurement and the best possible measurement that technology permits. User determines the privacy preference and the final area to be reached providing the

requested privacy. To enable the demanded privacy, obfuscation operators are introduced, grouped into two, basic obfuscation operators and operators formed as a combination of them. Basic obfuscation operators are Shift, Enlarge and Reduce. Shift operator shifts the center of the location measurement while Enlarge and Reduce are changing the radius of the measurement accordingly. Furthermore, combination of these operators can be obtained when these operators are applied as a sequence. The common characteristic of all these operators is that final area should have some overlapping parts with the initial area.

We used an attack algorithm proposed in [2] to attack the obfuscated data produced by the obfuscation techniques mentioned in the preceding paragraphs. In the scenario of the attack, adversary knows small number of trajectories and targets an unknown trajectory by using the dissimilarity matrix, which contains pairwise distances between trajectories. Besides, linear interpolation for the missing points was used. Linear interpolation is necessary to attack target with small number of known trajectories. It is because for $2n$ known trajectories, target is modeled as n points. In case n is smaller than the size of the trajectory, remaining points can be interpolated such that the interpolated points respect the matrix. Attack produces candidate trajectories for the target and by analyzing probability distribution of those candidates around an area, adversary can anticipate the presence or absence of the user in the chosen area.

For the attack algorithm mentioned above, we chose the sensitive areas to calculate the confidence of area to extract the presence information of the user. We observed that that above mentioned obfuscation methods are not adequate to preserve privacy of the user in POIs when the adversary of the attack chooses the radius of the areas as 500 or 1000 meters to calculate the confidence. Furthermore, those obfuscation techniques do not destroy the trajectories' linearity, which further helps the success of the attack.

The thesis is organized as follows: Chapter 2 gives information about the background and related work on our research area. In Chapter 3, we express the motivation for the thesis. In Chapter 4, the obfuscation method that we proposed is explained in detail. In Chapter 5, the alternative obfuscation technique is discussed. In Chapter 6, the applied attack method is analyzed while in Chapter 7, our exper-

imental setting is explained and results are discussed. In Chapter 8, we conclude the thesis and mention about future works.

Chapter 2

Background and Related Work

In this chapter, we will introduce background and related works in the fields of privacy in data mining in general, and then privacy preserving data mining, and privacy preserving data mining methods in spatio-temporal data which are anonymization and obfuscation.

2.1 Privacy in Data Mining

Privacy is defined as the significance of protecting personal data from public [4]. Revealing personal information is, unless user preferences states so, considered as privacy violation and all privacy preserving techniques are trying to prevent such cases. Increase in mobile devices and enhanced technology led to huge amount of spatio-temporal data. Thus, those collected vast data stimulated the privacy in spatio-temporal data. The important point for those collected data is that they should reveal information according to the preferences of people. If information goes beyond the choice of the individual, then privacy leak occurs. Furthermore, information through different channels may also result in privacy leak when they are combined and used for data mining which attempts extracting useful information from data. Such privacy leaks are presented in [5–7]. In the next section we present some of the available techniques to protect privacy.

2.2 Privacy Preserving Data Mining

As mentioned in the previous section, vast collected data have stimulated privacy preserving data mining. It is because of the aforementioned data that should be re-arranged according to the privacy demand of the users before using in data mining

techniques. Rearrangement of the data is accomplished through privacy preserving techniques. In [8], privacy is introduced in the field of data mining and later studies on this topic followed. One way of providing privacy is removing the attributes of data which lead to identification of a person. The authors of [9] propose a method which eliminates the particular data which may result in revealing the correlation rules in database. Data is still provided to the other parties but rules are kept private. Furthermore, in [10], a method that utilizes secure multi-party computation to cluster horizontally separated spatio-temporal data is proposed as a privacy preserving data mining technique.

2.3 Privacy in Spatio-Temporal Data

Increase in mobile devices which have GPS capability made location based services (LBS) prevalent as discussed in [1,11–14]. LBS, which are providing the service according to the user's position, increased the need for privacy techniques in spatio-temporal data. In this area, privacy preserving techniques can be categorized as anonymization and obfuscation.

One way of providing privacy for spatio-temporal data is anonymization which is discussed in [12, 15–19]. Among these works, in [16] and [17] anonymization of trajectories is targeted. However, in [20], it is shown that segments of trajectories can be used to reveal the remaining parts of trajectories. Furthermore, to prevent such a case, authors propose a method in which trajectories are considered as vertically partitioned and distributed across various parties where each party is trying to find out the rest of the trajectory. A suppression technique is introduced against such privacy leak.

Obfuscation is another approach to provide privacy for an individual which basically perturbs the data. However, it can be in the form of generalization as well. This technique is interest of research in [1,11,21]. In [1], authors propose obfuscation operators such as shift, enlarge, and reduce which are geometry based, not looking at the map information and each location point is treated as individually circular areas so path is not considered. However, in [21], authors propose a database level approach in which geographical information is also utilized while obfuscating. On the other hand, in [22] it is discussed that since the change pattern in the data can

be anticipated, obfuscation may not protect the privacy.

Chapter 3

Preliminaries and Thesis Motivation

In this chapter, we give the background information in Section 3.1 for the obfuscation method proposed in [1] and attack method presented in [2]. We used this obfuscation method and the method we designed to evaluate their effectiveness according to the attack method. After providing the background information for the thesis, we explain the thesis motivation in Section 3.2.

3.1 Preliminaries

3.1.1 Basic Concepts

In this section, basic concepts for the obfuscation operators discussed in Chapter 5 are explained.

Positions of users are represented as circular areas since it is not expected that sensing technologies would return exact location information.

Location Measurement

Location measurement for a user u, is denoted by $A_i = (x_i, y_i, r_i)$, calculated by location sensors, and following conditions hold:

1. $P((x_u, y_u) \in A_i) = 1$
2. $P((x_u, y_u) \in A)$, where $A = (x, y, \delta r) \subset A_i$ is the neighborhood of (x, y), is uniformly distributed

In these conditions, while first one states that a location measurement exactly contains the real position of the user, the second condition shows the real position of the user can be anywhere within A_i.

Relevance

Relevance is proposed as a metric of the privacy of location measurement and defined as $R_i = r_o^2/r_i^2$ for a location measurement $A_i = (x_i, y_i, r_i)$, where r_o is the radius of the area, the best that technology permits, while r_i is the radius of the A_i returned from the location sensors. As it is seen relevance is independent from the sensing technology and its value is in the range of $(0, 1)$, as it approaches to 1, the accuracy increases.

Location Privacy

In association with relevance, location privacy is $1 - R_i$, for the location measurement A_i. As relevance increases, location privacy decreases. Users can set their privacy preferences in terms of final relevance R_f, the relevance to be obtained, independent from the application context.

Accuracy Degradation

Accuracy degradation to be achieved is the ratio R_f/R_i, where R_i is the initial relevance and R_f is the final relevance set by the user (See Equation 3.1 and Figure 3.1). Obfuscation is done to achieve R_f, transforming the initial location measurement A_i to A_f in such a way that $P((x_u, y_u) \in A_f) > 0$ which implies A_i and A_f should be overlapping. Obfuscation is done only if $R_i > R_f$, otherwise the sensing technology already provides the requested privacy.

$$\lambda = \frac{R_f}{R_i} = \frac{(A_i \cap A_f)^2}{A_i A_f} \tag{3.1}$$

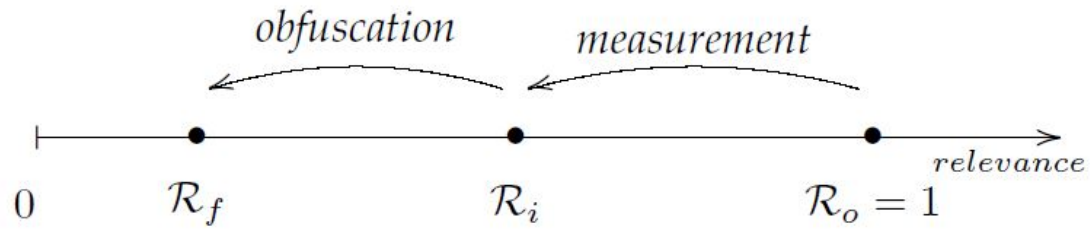


Figure 3.1: Accuracy degradation, taken from [1]

3.1.2 Attack Method

In [2], the authors propose a method which can identify any unknown trajectory, given the dissimilarity matrix and a small number of known trajectories. The method succeeds:

- Adversary is able to predict the presence or absence of an individual in a circular area with a certain confidence which is directly proportional to the radius of the area.
- Although a small number of trajectories are known, adversary is able to identify the presence with high confidence.

Algorithm works as follows; given r number of known trajectories, adversary is trying to reconstruct the target trajectory with p points. Each point has two unknown (latitude, longitude), therefore $2p$ equations are required to reconstruct the target, each pairwise distance between target and known trajectory forms an equation. Therefore, $2p$ equations correspond to $2p$ number of known trajectories. However, even though $r < 2p$, adversary can express the remaining $2p - r$ missing points in terms of the unknown points by using interpolation. Interpolation can be done in many ways, so each interpolation yields a different candidate trajectory. All candidate trajectories adapt the pairwise distances released through the dissimilarity matrix. Furthermore, without additional information, each candidate trajectory has equal probability to be the target. Since it is not practical to find all candidates, authors follow Monte Carlo approach; they run the algorithm many times, each with different interpolation and obtain a distribution to approximate the target. Then those candidate trajectories are used to infer absence or presence of an individual in a given area.

Dissimilarity Matrix

Trajectories are represented as a two dimensional vector $T = (p_1, ..., p_n)$ where point $p_i = (x_i, y_i)$ and they are produced by constant sampling rate, so time attribute is not used. As distance metric, Euclidean distance is used, the distance between

trajectories are defined as follows:

$$(T - T')_2 = \sqrt{\sum_{i=1}^{n} |p_i - p'_i|_2} \tag{3.2}$$

The technique uses Euclidean distance which is applicable to the trajectories of the same length. Dissimilarity matrix D is an $m \times m$ matrix for trajectories $ST = \{T^1, ..., T^m\}$, each value of the matrix is calculated as $D(i,j) = (T^i - T^j)_2$, thus each entry of the matrix corresponds to a pairwise distance between two trajectory and the matrix contains all pairwise distances between trajectories. An example is given below:

Data set	Distances
Trajectory 1:[(1,1)(2,2)(3,3)]	D(2,1)= $\sqrt{3}$
Trajectory 2:[(2,1)(3,2)(4,3)]	D(3,1)= $\sqrt{15}$
Trajectory 3:[(2,3)(3,4)(4,5)]	D(2,3)= $\sqrt{12}$

Dissimilarity Matrix			
Trajectory ID	1	2	3
1	0	$\sqrt{3}$	$\sqrt{15}$
2	-	0	$\sqrt{12}$
3	-	-	0

Table 3.1: Dissimilarity Matrix, taken from [2]

Problem Definition

It is stated that target trajectory T^r can be fully reconstructed if the number of known trajectories k has the following property, $k = 2n + 1$, where n is the number of points in trajectories. If $k < 2n + 1$, then there are infinite number of candidate trajectories, which are distance preserving, satisfying all pairwise distance conditions. Distance preserving trajectories cannot be distinguished from the target trajectory, however since it is not feasible to generate all possible candidate trajectories, following observation used to narrow the search space for the target:

- Consecutive data points in trajectory are not independent of each other. If a

point p_i is in sequence with another point p_{i+1}, it is most likely to be in the neighborhood of p_{i+1}.

This observation is the incentive for interpolated trajectories. A trajectory is modeled as including m points. If original trajectories contain more than m points, then the remaining points can be obtained through interpolation. m main points forms $m-1$ line segments and the number of interpolated points to include in each segment is determined by a set S.

The problem is defined as attacking a target trajectory, given a set of known trajectories of size $2k$, and a distance matrix, such that finding set of candidate trajectories which are distance preserving and remaining points out of k are interpolated.

Assumptions made by the method are as follows:

1. Trajectories are following a road network
2. Trajectories have a constant sampling rate
3. A small number of trajectories are known by adversary

Confidence-Based Attack

Confidence of a given area is the output of the attack algorithm which is the probability that target individual is present in the area. In order to do that, probability distribution around the point of interest is calculated and then later this value is used to infer with a certain confidence whether the target individual present or absent in that region. Algorithm works as follows:

1. A predefined number of iterations, distance preserving interpolated candidates are generated with ordered set of indices S which is randomly generated, and it yields each segment of candidate T^c has random number of interpolated points. All T^c are distance preserving and they have $\frac{k}{2}$ main points and the rest of the points $n - \frac{k}{2}$ are interpolated by utilizing S.
2. Then by using the candidate trajectories confidence of a given area A is calculated as $|CA|/|CT|$, where CT is the total number of candidate trajectories for the target, CA is the number of candidates passing through area A.

For example, as you can see in Figure 3.2, for the target T^r, four (T^{c1},T^{c2},T^{c3},T^{c4}) out of five trajectories passes through region shown as circle with radius r, therefore the confidence of area is 80%.

Candidate generation is done given the ordered set of indices S, and pairwise distance set. $n - \frac{k}{2}$ points are linearly interpolated. They are written in terms of the main points and by using the pairwise distances between trajectories, $\frac{k}{2}$ linear equations are solved, and each root of main points yields another distance preserving candidate trajectories.

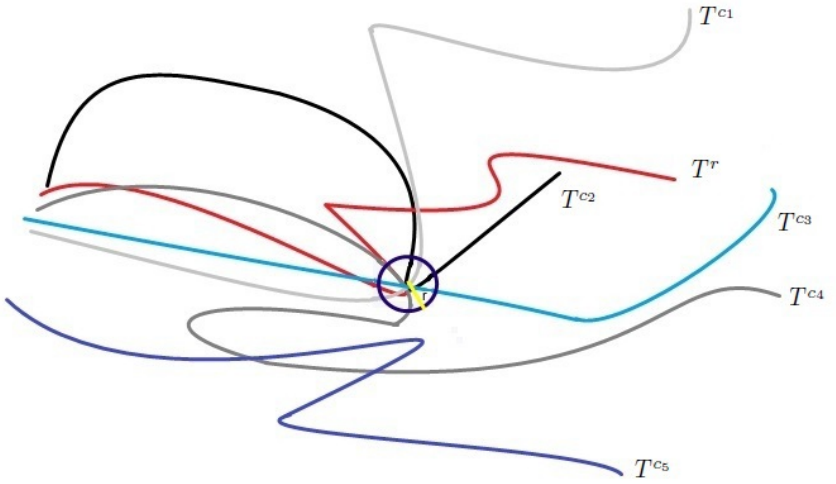


Figure 3.2: Confidence of area, taken from [2]

Higher the number of candidates, higher is the probability that the target trajectory is close to one of the candidates. Therefore size of the candidates is important for the attack in order to succeed.

Another criteria that is important for the success of the attack is sampling rate, as dependency between trajectory points increases, it makes possible the linear interpolation to be meaningful. Since the data used in our experiments has high sampling rate of 60 seconds, it is an advantage for the attack.

The attack method relies on the linear interpolation, so linearity of the data is an indicator of a successful attack.

Furthermore, in experiments, authors observe that as the number of known trajectories increases, adversary is able to obtain high confidence in smaller area while extracting the presence information about user.

3.2 Thesis Motivation

Increase in the availability of spatio-temporal data led to necessity of privacy preserving techniques. As discussed in the previous chapter, obfuscation is one of them. Obfuscation techniques changes the true location of the person or it generalizes the location, thus preserving the privacy. However, while evaluating privacy preserving techniques, we need to consider adversary behaviour and try to make the method resistant to such attacks.

We used two obfuscation techniques to protect privacy of trajectories in predefined sensitive areas. One of them is a method suggested in [1] and the other one is a method that we designed. We used those techniques for obfuscation of the data and ran the attack algorithm described in [2].

The method of [1] uses an adversary model in their work to evaluate the robustness of their obfuscation operators. If the de-obfuscation cannot be done successfully, those operators are said to be robust. If the adversary is not able to obtain a relevance higher than the relevance of the obfuscated area, de-obfuscation is interpreted as unsuccessful. The adversary is attacking by reducing or enlarging the radius of the obfuscated area to reduce the effect of obfuscation and obtain lower relevance. Their findings show that without the knowledge of the operator being used, the success rate of the adversary is below the 50%.

The method we designed is modeled such that it prevents adversary behaviour referring the map information and anticipating where the data is obfuscated. It is because the obfuscation is done only on sensitive data points. This protection is enabled by the method ensuring the obfuscated point to be on the road again. Besides, it is chosen as one of two points around the circle of next non-sensitive point and has the smallest distance from the original sensitive point. Therefore, the obfuscated point which is ensured as non-sensitive usually falls between the original sensitive point and the next non-sensitive point. Thus, trajectory is preserved.

However, when we use the attack scenario explained in [2], in which adversary knows a set of trajectories and trying to find out places where an unknown target trajectory is present or absent, given only a matrix containing pairwise distances between trajectories, these obfuscation techniques do not work. Adversary is still able to extract information about presence of target trajectory in sensitive areas even

though target's location points are obfuscated in sensitive areas. It is because when the area, in which presence is the concern, is chosen large enough that attacker is still able to conclude a person is present in that area. Those obfuscation methods do not help to protect his/her privacy since they do not scatter the obfuscated point enough to stay outside of that sensitive area. Furthermore, those approaches preserve the linearity of the trajectory after the obfuscation which further contributes to the success of the attack.

Chapter 4

Proposed Obfuscation Method

This chapter describes the obfuscation method that we designed and implemented. The method obfuscates data points of trajectories falling into predefined sensitive places by utilizing the map information.

4.1 Preparation of Data

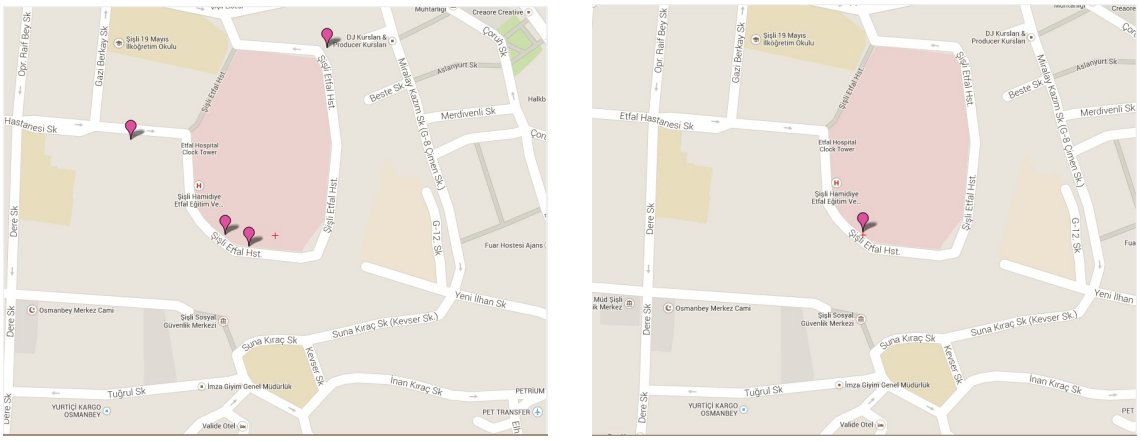
Preparation of data to obfuscate consists of two steps.

4.1.1 Elimination of Repetitive Points

First step of preparation is eliminating the repetitive points in trajectories, which is simply merging consecutive points in a trajectory that has the same latitude and longitude value. This is required for the Interpolation Attack to operate, since it calculates distance matrix and repetitive points in the trajectories cause singular matrix which is not invertible.

4.1.2 Merging Sensitive Points

The data points which are consecutive in the data (according to time) and falling into the same sensitive place are merged into one point which is the middle point according to their order in the data, in this way other sensitive points are eliminated.



(a) Before merging (b) After merging

Figure 4.1: Sensitive points before and after merging step

4.2 Sensitive Places

The sensitive places can be determined according to which places are considered as sensitive, it is a relative issue, while some people define cafes, bars as sensitive places, some might think hospitals, pharmacies to be sensitive. In this work, health related places such as hospitals, medical centers are defined as sensitive and they are considered as circular, specified by the latitude and longitude as their center, and radius value. While considering a point falling into a sensitive place, haversine formula is used to calculate distance between two points on a sphere, specified by latitude and longitude value (See Equation (4.1)), and GPS error is also taken into consideration. If the distance between data point and the center of the sensitive place is less than or equal to the radius of the sensitive place summed up with GPS error, the data point is considered as sensitive.

4.3 Method

The method only obfuscates the trajectory points that exist in predefined sensitive places (S). Therefore for each point, it is checked if the point is sensitive. By using Equation (4.1) distance between the sensitive location center and the point is found.

$$
\begin{aligned}
u &= \sin(\frac{lat2 - lat1}{2}) \\
v &= \sin(\frac{lon2 - lon1}{2}) \\
dist &= 2r_e \arcsin(\sqrt{u^2 + \cos(lat1)\cos(lat2)v^2})
\end{aligned}
\tag{4.1}
$$

r_e: Radius of earth

lat1, lon1: latitude, longitude of the sensitive place in radians

lat2, lon2: latitude and longitude of the trajectory point in radians

If the distance (dist) is smaller than the sum of radius of the sensitive location S_m and GPS error, this trajectory point is interpreted as sensitive and obfuscation is done according to following steps:

1. For each sensitive point in trajectory T_i we try to find the next non-sensitive data point (d_k)

2. When we find d_k a circle is formed around this point with radius calculated in Eqn. 4.2 which is the average neighboring distance of trajectories. Around the circle, candidate points to replace the sensitive points are formed such that all are equally separated by angle, which is determined by the predefined number of candidate points. Finding latitude and longitude of candidate points is calculated in Eqn. 4.3

$$
\sum_{\forall T_i \in T} \sum_{k=0}^{n-2} dist(d_{k+1}, d_k)
\tag{4.2}
$$

dist: distance calculated between two points according to Eqn. 4.1

n: number of data points in a trajectory (one point contains both latitude and longitude value, all trajectories have the same number of data points)

$$
\begin{aligned}
lat &= \arcsin(\sin(lat_1)\cos(d) + \cos(lat_1)\sin(d)\cos(\theta)) \\
dlon &= \arctan(\frac{\sin(\theta)\sin(d)\cos(lat_1)}{\cos(d) - \sin(lat_1)\sin(lat)}) \\
lon &= ((lon_1 - dlon + \pi) \bmod 2\pi) - \pi
\end{aligned}
\tag{4.3}
$$

θ: Angle between the non-sensitive point and the candidate point, if it is zero, then longitude does not change above lon calculation is not used

lat, lon: Calculated latitude, longitude value of the candidate

d: Distance, which is the radius of the circle in our case. It is in radians so

the radius is divided by the earth radius

lat_1, lon_1: Latitude, longitude of the non-sensitive point which is used as the center of the circle

Each candidate is produced with different θ value which is starting from 0 and increased by 2π/(number of candidates) for each candidate until 2π. Among those candidates, sensitive ones are eliminated.

3. All candidate points are mapped to the nearest road as we are dealing with vehicle trajectories.

4. If the number of candidate points is less than n (defined as 2 in our case), then the radius of the circle is increased by a predefined amount (10 meters in our case) and the procedure is repeated from 2 to 4 until at least n number of non-sensitive points are found around the circle mapped to the road.

5. Those candidate points are sorted according to their road distance from the original sensitive point.

6. Among the candidate points, n points having the smallest distance is determined and one of these n points is selected randomly to replace the sensitive point.

7. If the last point of the trajectory is sensitive, there isn't any next non-sensitive point, then the same procedure is applied from the 2, however the circle is formed around this sensitive point instead of the next non-sensitive one.

An example of the obfuscation method is shown in Figure 4.2. Candidates around the next non-sensitive point formerly form a circle, and then those points are mapped into the nearest road around them. These points are sorted according to their distance. One of the n points (2 is chosen in our experiments) having smallest distance is chosen randomly to replace the sensitive point.

Figure 4.2: Road mapped candidates around next non-sensitive point

Pink Pinpoint: Sensitive trajectory point

Green Pinpoint: Candidate point formed to replace sensitive point

Purple Pinpoint: Non-sensitive trajectory point

Orange Pinpoint: Chosen point to replace sensitive point

Chapter 5

A State of the Art Obfuscation Method

This chapter is about the obfuscation method explained in [1]. We explain obfuscation operators proposed by the authors, discuss their effectiveness for the attack method that we used in this thesis, and then we compare this method with our obfuscation method explained in Chapter 4.

5.1 Obfuscation Operators

All obfuscation operators takes A_i,R_f and R_i as input and gives an output which is the obfuscated area with relevance R_f. Next section defines basic operators which achieve obfuscation either by changing the radius or shifting the center. Furthermore, we discuss those operators based on their effectiveness when we use this operator in a setting that sensitive places are defined and the goal is to hide the location information falling into a sensitive place. Later, combination of these operators are explained. At last, we compared the method we designed and explained in Chapter 4 with the method explained in this chapter.

5.1.1 Basic Obfuscation Operators

Enlarge (E)

Produces an obfuscated area which has radius $r_f > r_i$. This operator decreases the probability that the real position of the user is a particular point within the neighborhood of A_f, while the real position remains in A_f (See Figure 5.2). Final radius is calculated by following Equation (3.1)

$$\frac{R_f}{R_i} = \frac{(A_i \cap A_f)^2}{A_i A_f} = \frac{A_i}{A_f} = \frac{r_i^2}{r_f^2} \tag{5.1}$$

$$r_f = r_i \sqrt{\frac{R_i}{R_f}} \tag{5.2}$$

The effect can be seen in Figure 5.1, A_f is shown by red circle while A_i is demonstrated by blue circle.

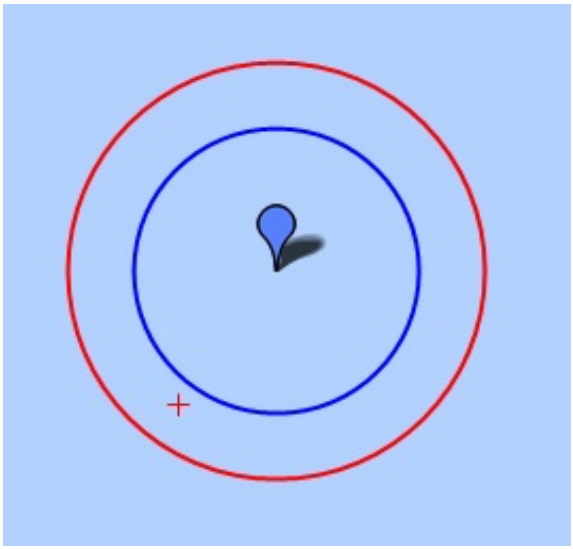


Figure 5.1: Enlarge (E) Operator

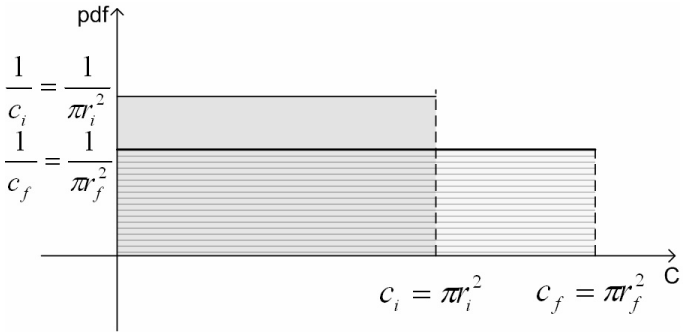


Figure 5.2: pdf of Enlarge (E) Operator

When we use this operator in a setting mentioned in Section 5.1, this operator does not operate well, because if a location measurement overlaps with the sensitive area, obfuscation based on enlargement does not remove its overlapping. In fact, the overlapping area usually gets bigger, so it is still overlapping with the place that presence of the individual is supposed to be hidden, and it is inferred that the individual is in the sensitive place (See Figure 5.3). Therefore in the attack that is explained on the Chapter 6, this operator wouldn't serve as a protection of sensitive data.

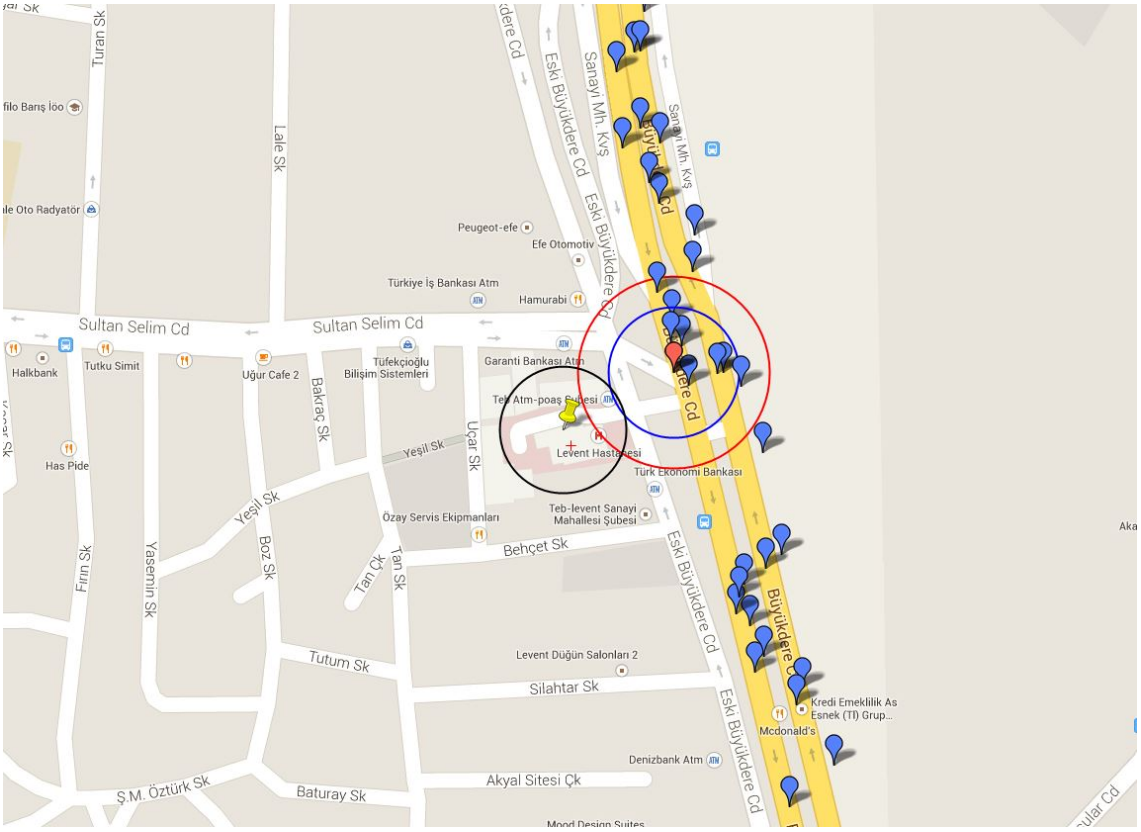


Figure 5.3: Enlarge (E) operator, bad case

Reduce (R)

Produces an obfuscated area which has radius $r_f < r_i$. Although this operator may not seem right as an obfuscation technique, it decreases the probability that the real position falls within the region A_f while the probability density function's(pdf) value remains the same (See Figure 5.5). Final radius is calculated by following Equation 3.1

$$\frac{R_f}{R_i} = \frac{(A_i \cap A_f)^2}{A_i A_f} = \frac{A_f}{A_i} = \frac{r_f^2}{r_i^2} \tag{5.3}$$

$$r_f = r_i \sqrt{\frac{R_f}{R_i}} \tag{5.4}$$

The effect can be seen on Figure 5.4, A_f is shown by red circle while A_i is demonstrated by blue circle.

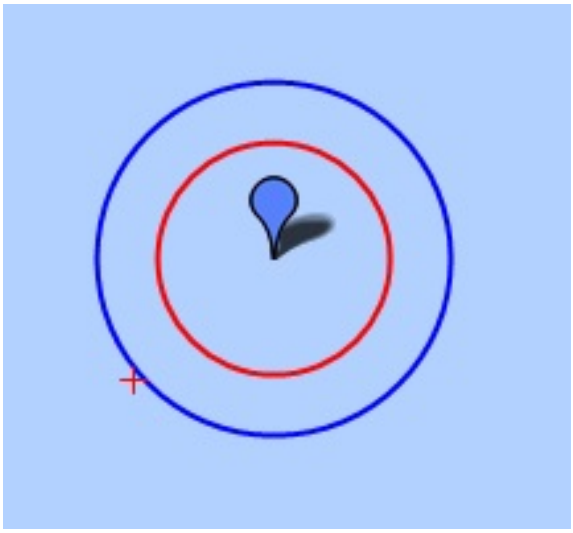


Figure 5.4: Reduce (R) Operator

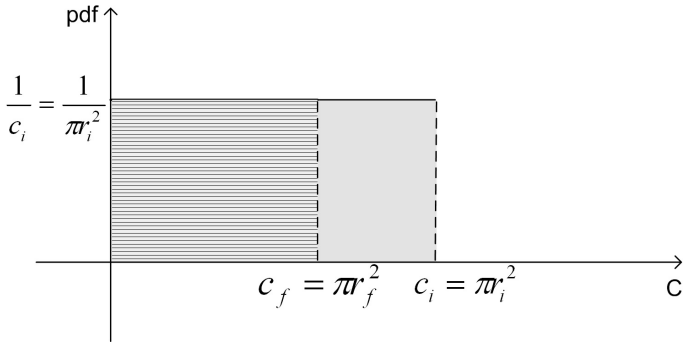


Figure 5.5: pdf of Reduce (R) Operator

This operator could work well in the context explained in Section 5.1. It occurs when the original location measurement extends over a sensitive place, and it is obfuscated with R operator such that no overlapping part with the sensitive place exists anymore as is seen on Figure 5.6) where a sensitive place is shown by a black circle, centered on yellow pinpoint.

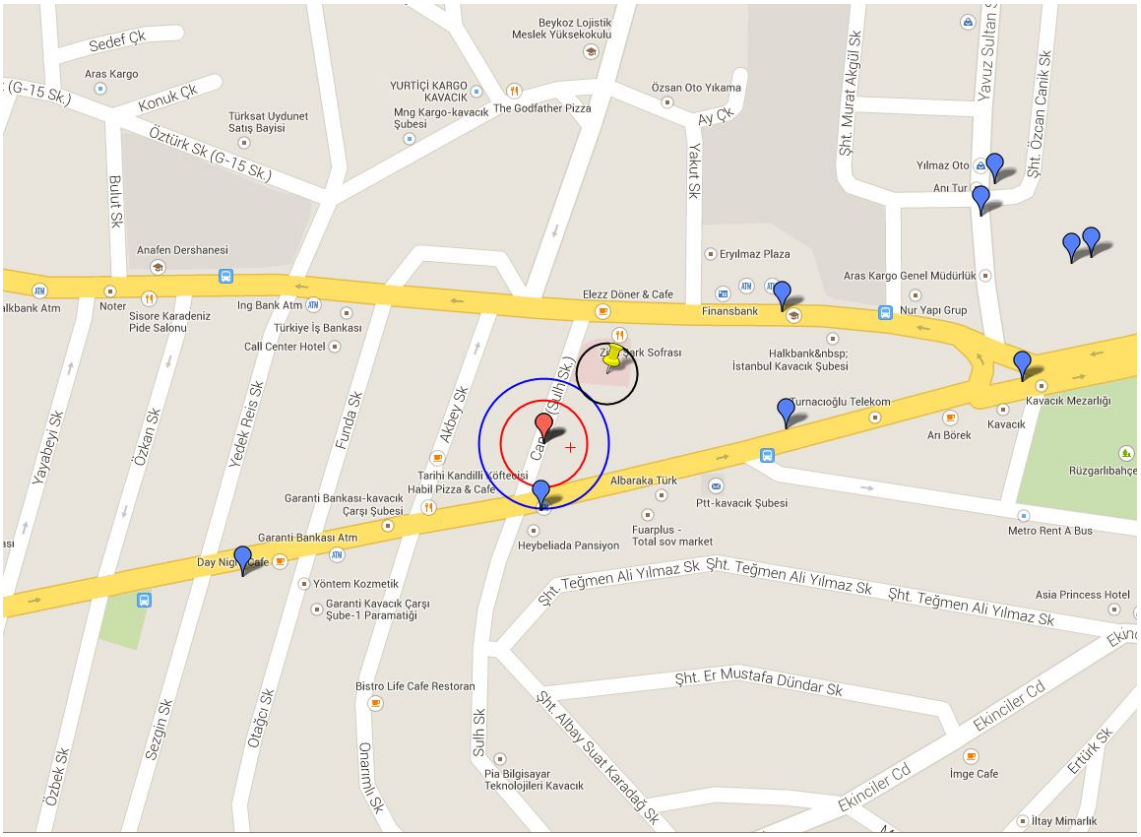


Figure 5.6: Reduce (R) operator, good case

However, there can be a situation when R operator does not help conserving the sensitive data point. As you can see on Figure 5.7, original data point resides in the sensitive place, R operator, reduces its radius but the location measurement of the data point still overlaps with the sensitive area.

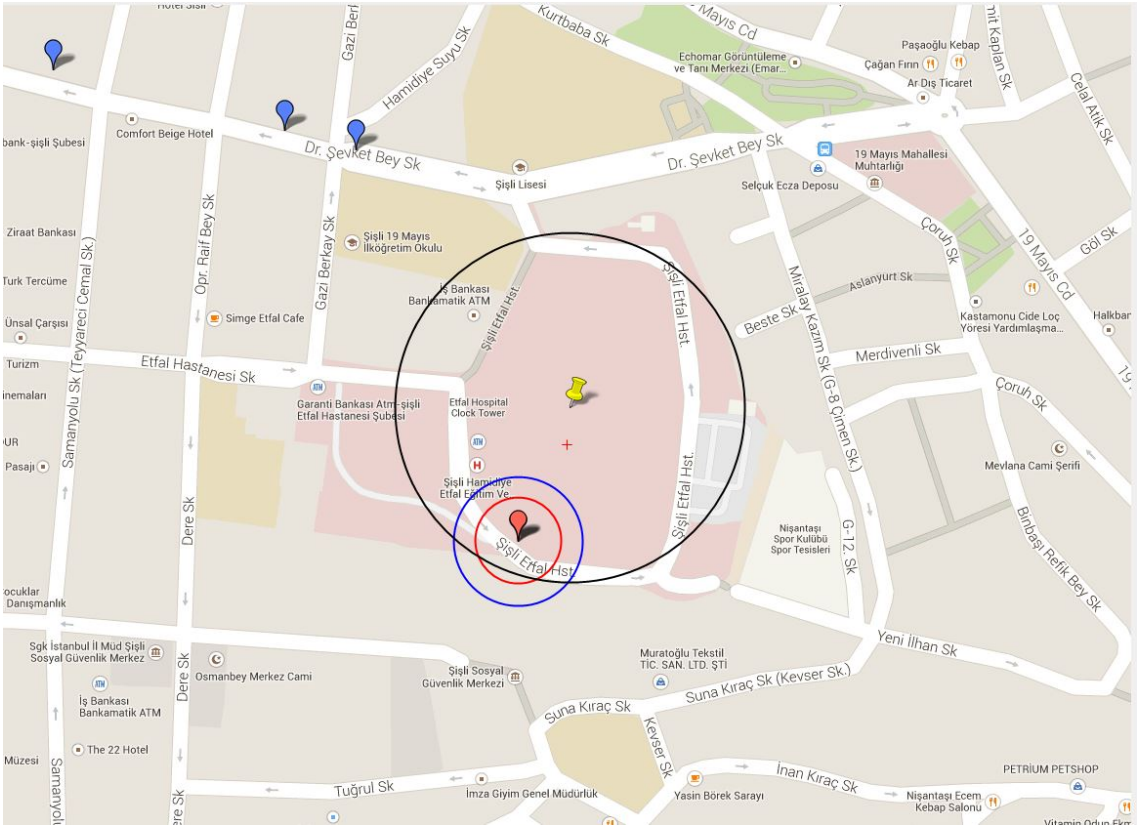


Figure 5.7: Reduce (R) operator, bad case

Shift (S)

Unlike the previous operators, shift operator does not change the radius, instead it changes the center of the area such that $(x_f, y_f) = (x_i + d\sin(\theta), y_i + d\cos(\theta))$, where $d \in (0, 2r_i]$ is the distance between the center of A_i and A_f, and θ is the rotation angle. This operator decreases both the probability that the real user is in the neighborhood of A_f and the probability that the real position is contained within A_f. θ is generated randomly and d is found by solving the following equations:
Since A_i and A_f have the same area, from Equation 3.1

$$A_i \cap A_f = \pi r_i^2 \sqrt{\frac{R_f}{R_i}} \tag{5.5}$$

When the term $A_i \cap A_f$ is interpreted by the distance d between the centers, following equations formed, where σ and γ are central angles of circular areas A_i and A_f, $\lambda = R_f/R_i$

$$\left[\frac{\sigma}{2}r_i^2 - \frac{r_i^2}{2}\sin\sigma\right] + \left[\frac{\gamma}{2}r_f^2 - \frac{r_f^2}{2}\sin\gamma\right] = \sqrt{\lambda}\pi r_i r_f$$
$$d = r_i\cos\frac{\sigma}{2} + r_f\cos\frac{\gamma}{2} \tag{5.6}$$
$$r_i\sin\frac{\sigma}{2} = r_f\sin\frac{\gamma}{2}$$

Since areas have the same radius value $r_i = r_f$, those equations simplify as,

$$\sigma - \sin\sigma = \sqrt{\lambda}\pi$$
$$d = 2r_i\cos\frac{\sigma}{2} \tag{5.7}$$

Solving the Equation 5.7 gives the distance d, and the other unknown θ is chosen randomly to calculate obfuscated area A_f.

See Figure 5.8, A_f is shown by red color while A_i is demonstrated by blue.

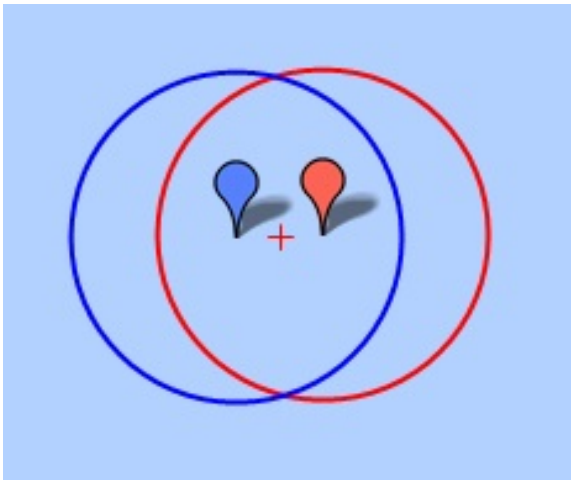


Figure 5.8: Shift (S) Operator

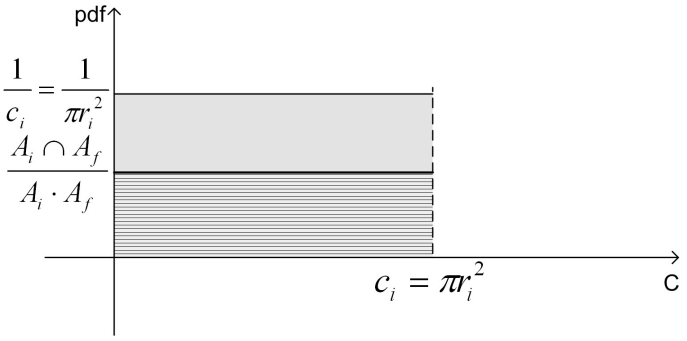


Figure 5.9: pdf of Shift (S) Operator

This operator may not result in a desired obfuscation according to a sensitive place, it is because we have high precision of location measurement as presented in

Section 7.4, so S operator does not perturb data enough to protect the sensitive data point. Furthermore, rotation angle is chosen randomly, sensitive place's location is not considered. Therefore it may result in such a case in Figure 5.10. The S operator does not serve as a privacy measure for this particular data. Therefore, in the attack of Chapter 6, this operator is not expected to succeed.

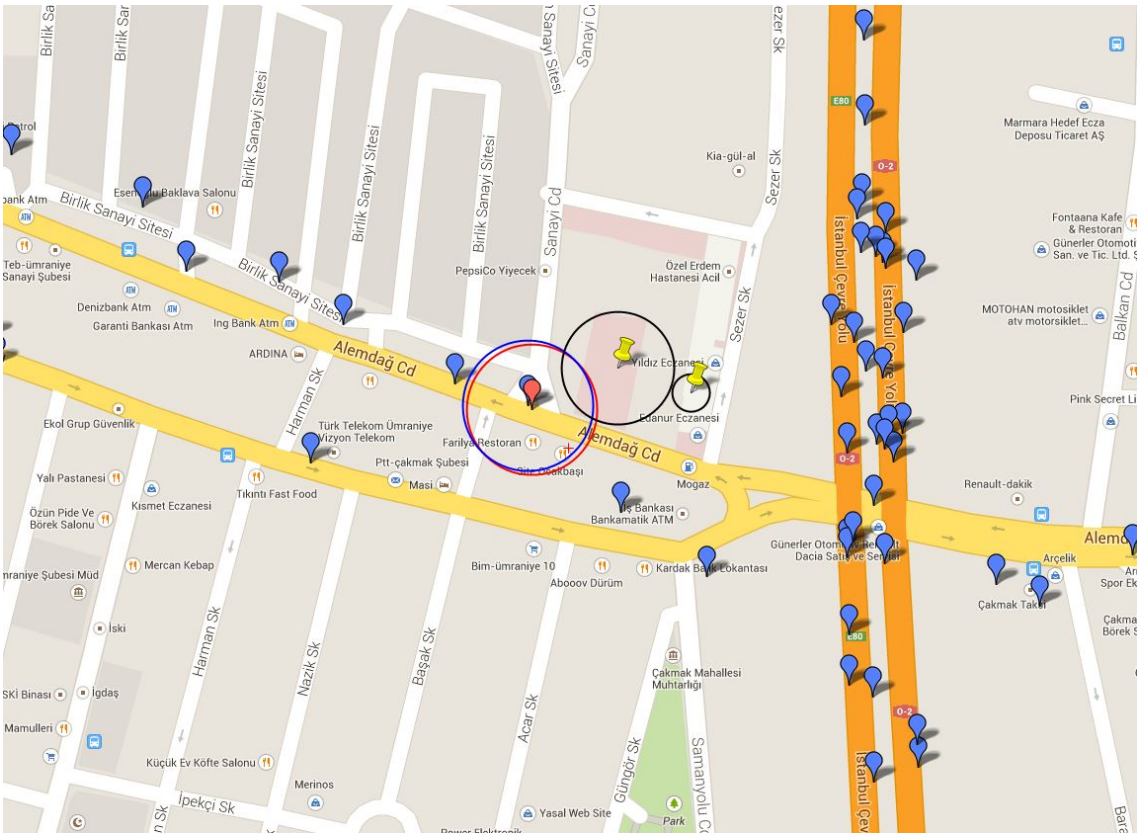


Figure 5.10: Shift (S) operator, bad case

5.1.2 Combination of the Basic Obfuscation Operators

Combination of basic obfuscation operators is when two of them is applied in sequence. Authors state that it is not necessary to combine more than two operators since an area $A_1 = (x_1, y_1, r_1)$ can be converted to another area $A_2 = (x_2, y_2, r_2)$ by only applying two operators:

- Shifting the center
- Enlarging or reducing the radius

The important point to note while combining operators is that the relation between final area A' and the initial area A_i is $A' \cap A_i \neq \emptyset$. Furthermore, while combining

operators, relevance is decreased gradually. First operator decreases the relevance from R_i to R_m which is chosen randomly between R_i and R_f and produces area $A_m = (x_m, y_m, r_m)$, second operator decreases relevance R_m to R_f which is the final relevance to be achieved at the end of obfuscation according to user privacy preference and produces the final area $A_f = (x_f, y_f, r_f)$(See Figure 5.11). In following sections operators used in this work are explained.

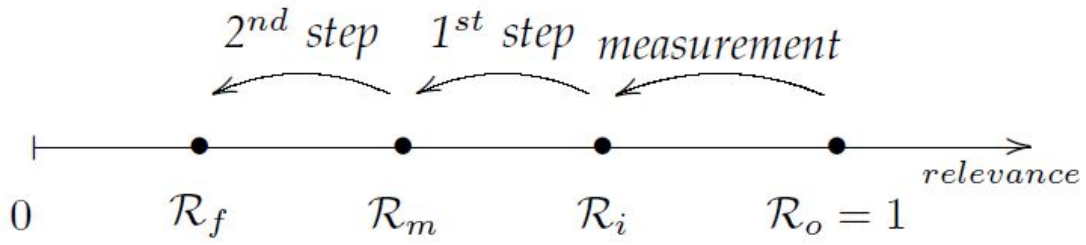


Figure 5.11: Relevance while combining operators, taken from [1]

Shift and Enlarge (SE) Operator

First Step is Shift Operator which is done by applying the operator explained in 5.1.1. Input is $A_i = (x_i, y_i, r_i)$, output is $A_m = (x_m, y_m, r_m)$. Calculation of x_m,y_m and r_m is shown on Table 5.1. In this calculation, d is found by Equation (5.7).

x_m	y_m	r_m
$x_i + d\sin\theta$	$y_i + d\sin\theta$	r_i

Table 5.1: Shift Operator when used as first step in combination of operators

Second Step is Enlarge Operator which is done by applying the Enlarge Operator mentioned in 5.1.1. In this part, whether the initial area A_i is fully included in the final area A_f or they are partially overlapped, these two cases are treated differently since authors states that these two cases have different behaviours when analyzed against an attack to eliminate obfuscation effects. Input is $A_m = (x_m, y_m, r_m)$, which is the output of the first step, output is $A_f = (x_f, y_m, r_f)$. Calculation of x_f,y_f and r_f for partial overlapping case is shown below. In this calculation, r_f is found by Equation (5.6).

x_f	y_f	r_f
x_m	y_m	$> r_m$

Table 5.2: Enlarge Operator when used as second step in combination of operators, partial overlapping case

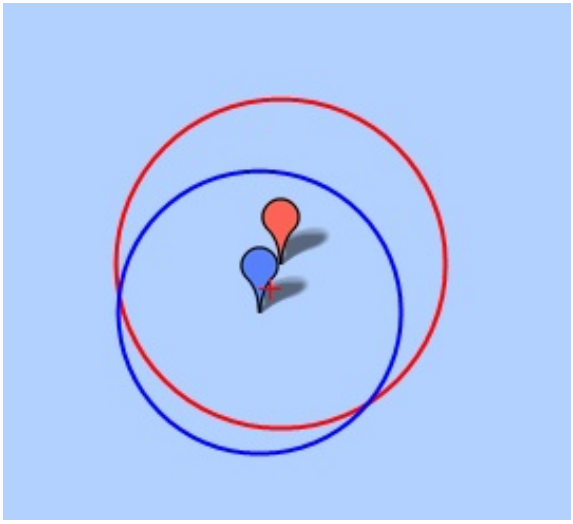


Figure 5.12: SE Operator, partial overlapping case

Calculation of x_f,y_f and r_f for inclusion case is shown below.

x_f	y_f	r_f
x_m	y_m	$r_m\sqrt{\frac{R_i}{R_f}}$

Table 5.3: Enlarge Operator when used as second step in combination of operators, inclusion case

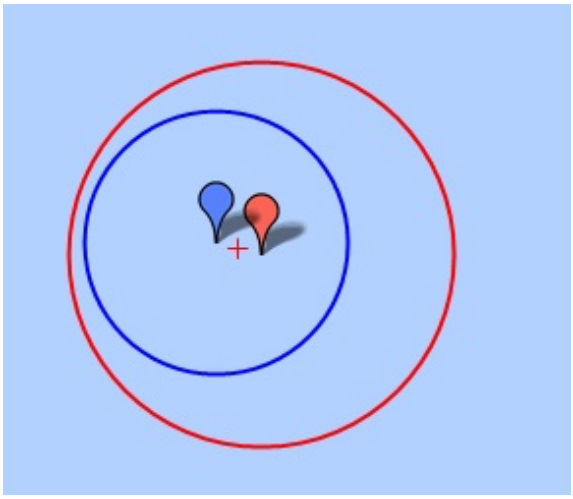


Figure 5.13: SE Operator, inclusion case

Shift and Reduce (SR) Operator

First Step is Shift Operator which is done by applying the operator explained in 5.1.1. Input is $A_i = (x_i, y_i, r_i)$, output is $A_m = (x_m, y_m, r_m)$. Calculation of x_m,y_m and r_m is shown on Table 5.1.

Second Step is Reduce Operator which is done by applying the Reduce Operator mentioned in 5.1.1. Input is $A_m = (x_m, y_m, r_m)$, which is the output of the first step, output is $A_f = (x_f, y_m, r_f)$. Calculation of x_f,y_f and r_f for inclusion case is shown below.

x_f	y_f	r_f
x_m	y_m	$r_m\sqrt{\frac{R_f}{R_i}}$

Table 5.4: Reduce Operator when used as second step in combination of operators, inclusion case

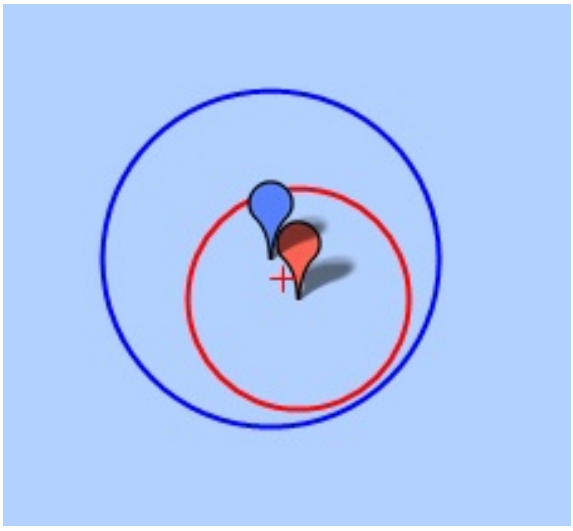


Figure 5.14: SR Operator, inclusion case

We used operators explained in this chapter for obfuscation of our data. Our setting and results are explained in Chapter 7.

5.2 Comparison of Obfuscation Methods

In this work, we used two obfuscation methods, the method explained in this chapter and the method introduced in Chapter 4 that we designed. In this section their differences are examined.

The method explained in Chapter 4, perturbs the data such that original point does not necessarily overlap with the obfuscated point, whereas the method explained in this chapter ensures the final location measurement to overlap with the original point so it does not spoil the data entirely.

Our method ensures that the obfuscated point does not fall into the sensitive place anymore. However the method explained in this chapter does not guarantee that the obfuscated point will not reside in the sensitive place. We showed those cases in figures of Section 5.1.1.

In the introduced method of Chapter 4, obfuscation is done such that the trajectory is not ruined, because the places where data points are out of the trajectory can be used by the adversary to attack by just looking at the map and infer that the data point actually falling into a sensitive place and it is perturbed. Therefore this approach can be protective based on attacks involving looking at the map, but it

cannot protect the sensitive data when linear interpolation based attack is applied as mentioned on Chapter 6.

On the other hand, the method mentioned on [1] treats location points individually, not considering as if they are connected through a path. This method may reveal sensitive information when only sensitive data points are obfuscated such that change in the precision of points may indicate sensitivity of the data point. Furthermore, the perturbation including S operator can result in the obfuscated point to be out of the trajectory if the precision of location measurement is low. However, the data used in this work has high precision as mentioned in Section 7.4.

Another aspect that differs in those approaches is utilizing map information. Introduced method in the previous chapter, takes advantage of it and candidate points are mapped to the road and one of the two points having the shortest road distance from original point to the candidate point is chosen randomly. Therefore, it can be said that this method is highly dependent on the map information whereas the method of [1] does not take it into account, operators including S, E may produce areas where a vehicle cannot reside in such as lake, car banned places or any other place in which roads are not contained.

In addition to that, the method of Chapter 4 utilizes background information of the data. It produces obfuscated points ensuring they are still following the same trajectory. However this is not the case for the method explained in this chapter. It does not take into account the background information of the data. If it is used on data with low precision of location measurement, the resulting obfuscated points that are out of trajectory can be used on the map to infer an individual is in a sensitive place.

Chapter 6

Attack Method

In this chapter, effectiveness of obfuscation approaches mentioned in Chapter 4 and Chapter 5 are discussed when the attack method proposed in [2] is used.

Preliminaries for this attack method are explained in Chapter 3.

The method of Chapter 4, ensures the obfuscated point is on the road, and it is the candidate point having one of the two smallest distances from original sensitive point, and is located around the next non-sensitive point. Purpose of such obfuscation is not to spoil the trajectory, but it further helps keeping the linearity as seen in the Figure 6.1, where red line indicates the original trajectory, blue line is the obfuscated trajectory, the direction of the movement is indicated with the arrow. As is seen, the trajectory's linearity is preserved making it susceptible to the attack method.

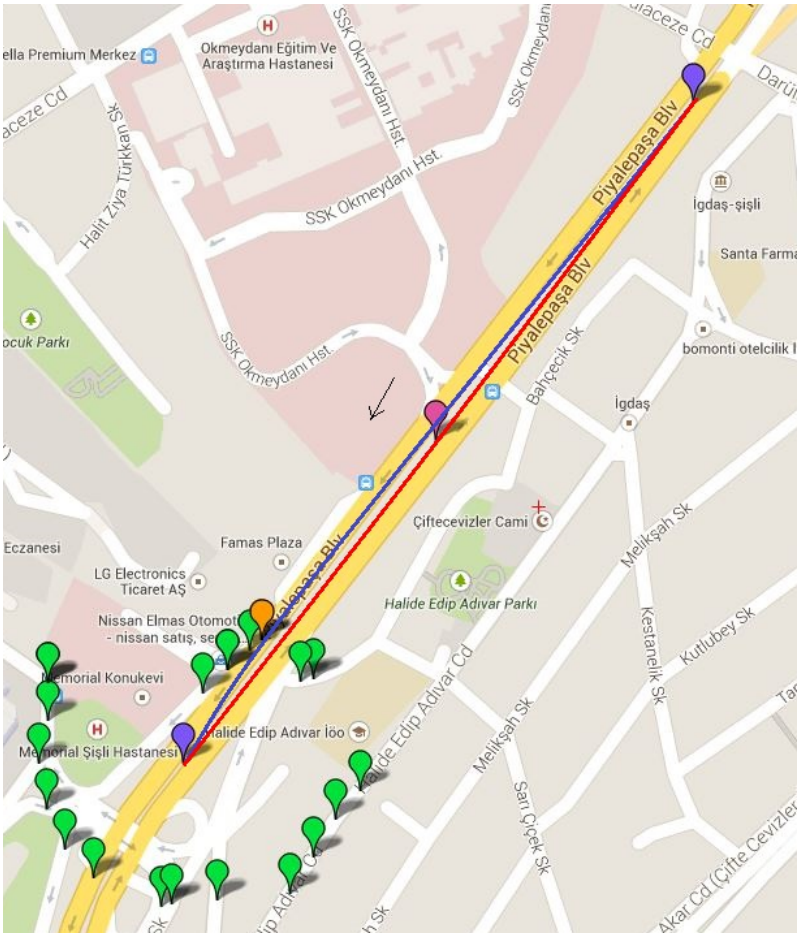


Figure 6.1: Linear approach of the obfuscation method introduced in Chapter 4

Furthermore, obfuscation operators explained in Chapter 5 does not perturb the trajectory, thus keeping the linearity. When Operators E and R are used individually, they only change the radius of the location point. If S operator included in the obfuscation, the location area's center is shifted. Since we initially have high precision of location measurement and resulting obfuscated areas should overlap with the initial location measurement, we didn't see any shift of the center that destroy trajectory fully, as we examined the obfuscated data. We can see an example in Figure 6.2, where green line shows original path and orange line shows the path after the obfuscation. Although data point is shifted, trajectory is slightly moved and linearity is preserved. Thus, it can be concluded that the attack method could succeed on this obfuscation technique.

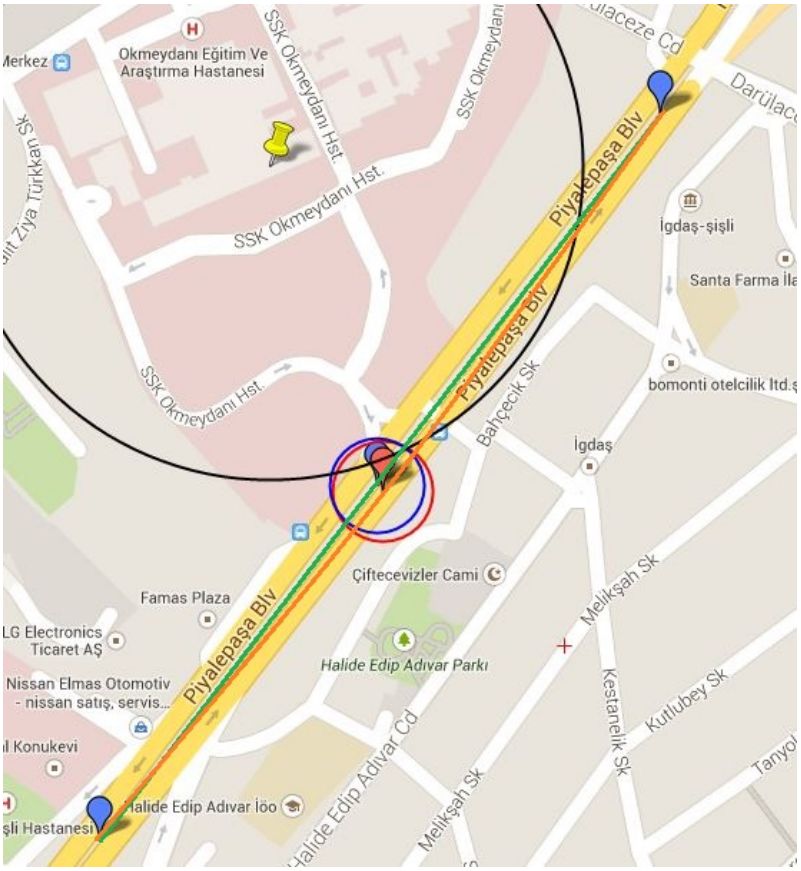


Figure 6.2: Linear approach of the obfuscation method introduced in Chapter 5

Moreover, the radius values of 500 meters or 1000 meters are chosen for the area of which confidence of area to be calculated. From the attacker point of view, this is accurate enough to conclude that the target trajectory actually appears in this area. To be enduring for such an attack, the obfuscation method should ensure that the obfuscated points are 500/1000 meters away from the sensitive locations. However methods explained in Chapter 4 and Chapter 5 does not work this way.

In our obfuscation method, we only ensure that the obfuscated point does not fall into the sensitive place, which has a 44 meters radius on average. Our method takes into account the direction of the movement and preserving the trajectory. Thus, the obfuscated point usually falls into between the original sensitive point and the next non-sensitive point (See Figure 6.3). Average neighboring distance for our data is calculated to be 90 meters as mentioned in Section 7.4. Therefore, our method does not protect the sensitive data when such big radius values are chosen to calculate the confidence of area.

Figure 6.3: Distance between center of sensitive area and the obfuscated point with circle around with radius of GPS error

Pink Pinpoint: Sensitive trajectory point

Green Pinpoint: Candidate point to replace sensitive point

Purple Pinpoint: Non-sensitive trajectory point

Orange Pinpoint: Chosen point to replace sensitive point

The method explained in Chapter 5 does not even guarantee that the obfuscated point does not reside in the sensitive area (See Figures 5.3, 5.7, 5.10). Even if the produced obfuscated circular area does not overlap with the sensitive place, it is not expected that the obfuscated area would be 500 meters away from the sensitive location center. It is because the final area produced by the operators should have some overlapping part with the initial area. In our context, initial area has radius (r_i) of 30 meters as mentioned in Section 7.4. In order to be sensitive, this initial location measurement should have overlapping parts with the sensitive area A_s (we only obfuscate when the initial location measurement is sensitive). As is seen in Figure 6.4, the case, where the obfuscated final area is the farthest from the center

of the sensitive area, is obtained by the S operator or the SE, partial overlapping operator. In those cases, the distance can be at most $r_s + 2r_i$ where r_s is the radius of the sensitive area. In our setting, r_i is 30 meters and average of r_s is 44 meters. Therefore, on average the obfuscated area is 104 meters away from the midpoint of sensitive place. This indicates those obfuscation operators are not adequate to protect the data when radius of 500 meters or more is used to calculate the confidence of area.

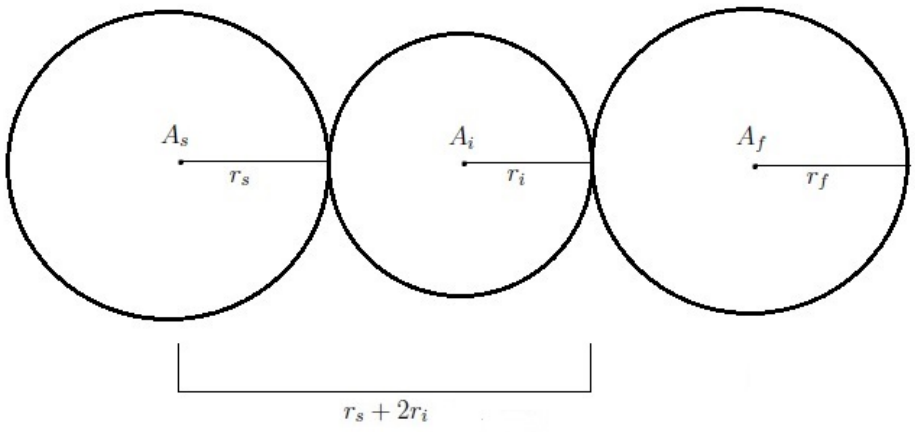


Figure 6.4: Distance calculation for the case where the obfuscated final area is the farthest from the center of the sensitive area

Chapter 7

Implementation and Experimental Results

In this chapter, firstly, we explain the format used in our experiments for trajectories and sensitive data. In Section 7.3, we give information about the tool that we have implemented. Then, in Section 7.4, the setting for our experiments is explained. The parameters used in obfuscation methods, the approach we followed in obfuscation and our confidence of area calculation are explained. At last, results are discussed in Section 7.5 according to the outputs of experiments.

7.1 Trajectory Data

We used vehicles' trajectory data tracked by GPS for 30 days. Initially data had attributes such as vehicle id, position time stamp, latitude, longitude, direction and speed as seen in Table 7.1. Data attributes such as time stamp, direction, speed are not used in our context. We further processed the data such that if the vehicle does not move more than 2 minutes, we started a new trajectory, thus trajectory ids formed and vehicle ids are not used. Then we realized there are some repetitive points in the data, we eliminated them as mentioned in Section 4.1.1, and then consecutive sensitive data points are merged as explained in Section 4.1.2. The size of the trajectories did not match, which is necessary for the attack to operate, therefore we determined the size of 125 points for each trajectory and the rest of the data has been removed. The format of the data used in obfuscation methods explained in this work is in Table 7.2. Each point is represented by point ID which is unique in all data, trajectory ID which is the trajectory the point belongs to, trajectory point ID, the point's ID in the trajectory and the point's latitude and longitude values. At the end of the operations we had 183 trajectories each having

125 points.

Vehicle ID	Time Stamp	Latitude	Longitude	Direction	Speed
38347	07/10/2013 17:43:39	40.987517	29.023072	GUNEY	0
52689	07/10/2013 17:43:09	40.978217	29.093	BATI	28
35882	07/10/2013 17:42:55	41.08417	28.97802	GUNEYDOGU	0
38346	07/10/2013 17:42:46	40.967903	29.08698	GUNEYBATI	0

Table 7.1: Example location data of company vehichles

Point ID	Trajectory ID	Trajectory Point ID	Latitude	Longitude
1	0	1	41.00311	29.022313
2	0	2	41.005005	29.029652
3	0	3	41.000322	29.05249
4	0	4	40.991113	29.077215

Table 7.2: Data format used in obfuscation methods

7.2 Sensitive Data

Sensitive data used in experiments described in Section 4.2. We extracted 45 health related places in Istanbul such as hospitals, medical centers etc. Each sensitive place is considered as circular and represented by the latitude and longitude value and the radius as seen in Table 7.3. Radii of sensitive places are calculated individually.

Latitude	Longitude	Radius
41.2903	27.9984	25.9605
41.2895	28.0016	24.2604
41.294	28.0051	27.6763
41.2859	27.9968	30.2208
41.2851	28.0012	29.8698

Table 7.3: Sensitive data format

7.3 Map Based Obfuscation and Visualization Tool

We implemented a tool to run the obfuscation algorithms mentioned in Chapter 5 and Chapter 4 (See Figure 7.1)

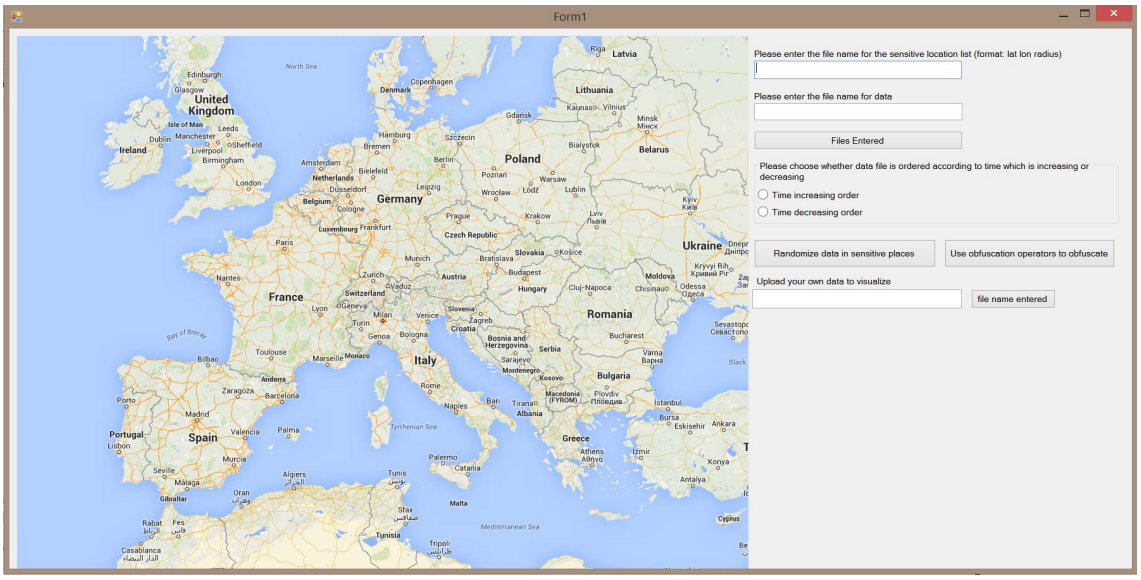


Figure 7.1: Tool

In this tool, we can visualize the data produced by obfuscation algorithms. Map figures used in this work are produced by this tool. User can use this tool either to visualize his/her data or to obfuscate his/her data and visualize it at the same time.

If the user wants to obfuscate his/her own data, he/she enters the file name for sensitive location and then the file name for the data to obfuscate (in formats shown in Table 7.2 and Table 7.3) and presses "Files Entered" button. Meanwhile

file checks are done whether they exist or not, user is then given choices to choose how his/her data is ordered according to time, whether the time is increasing as you move down in the list or is decreasing. We didn't include time property but still the order of the trajectories is important as it is explained in Chapter 4, the obfuscation of a sensitive point is done according to its next non-sensitive point according to time. This option is put on the tool to be more flexible to operate on any data. However, ordering is not important for the obfuscation method explained in Chapter 5. Then the user chooses between obfuscation methods in Chapters 4 and 5. The user presses the button according to which operation they want; according to which the corresponding algorithm starts to run.

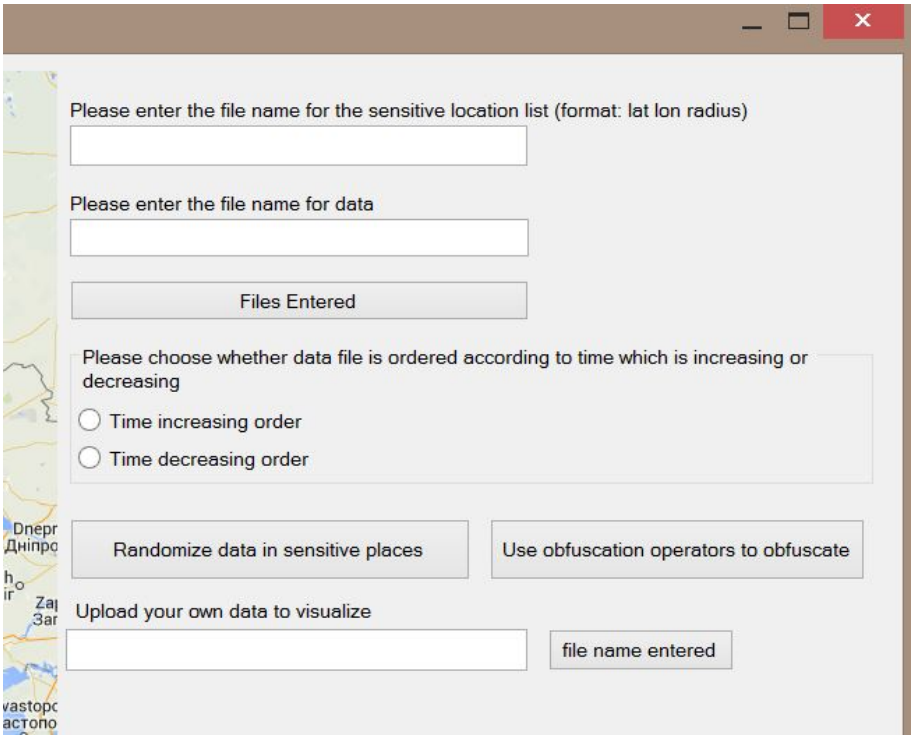


Figure 7.2: Tool close look

When calculations are done by the program, pinpoints on the map emerge. Then user can zoom in to see further details in the data.

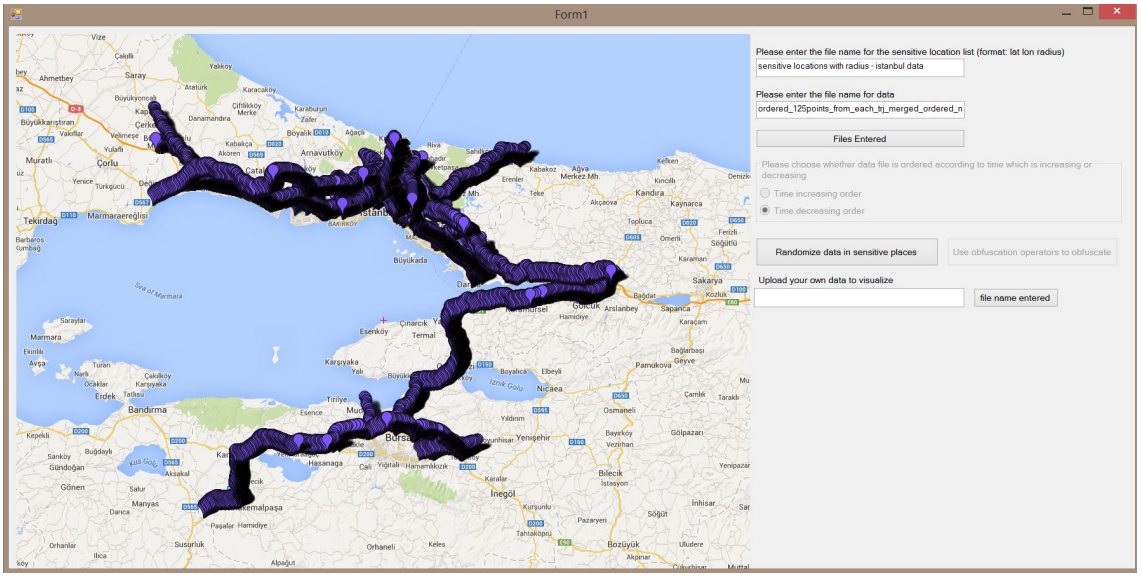


Figure 7.3: Tool runs on data to obfuscate

Data after obfuscation of Chapter 4 can be seen on Figure 7.4. Non-sensitive trajectory points are shown with purple pinpoints, while sensitive ones are pink, candidate points are green and the chosen points to replace the sensitive ones appears as orange pinpoints. Furthermore, it is possible to access the information of trajectory id and the order of the point in trajectory when the mouse is over a pinpoint. In this way it is possible to analyze the obfuscation.

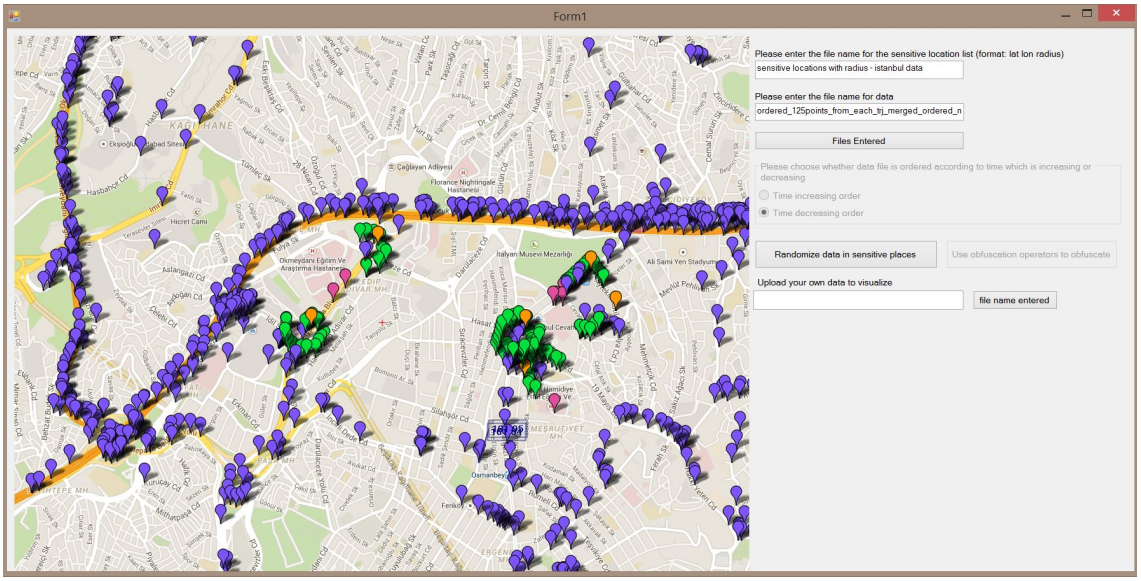


Figure 7.4: Trajectories visualized after obfuscation method of Chapter 4

The resulting location data as the output of the obfuscation method in Chapter 5 can be seen in Figure 7.5 and data in more detail on Figure 7.6. Since this

obfuscation method has operators such as E, R which are changing the radius of the circular area, obfuscated location circles are indicated by red pinpoint and red circle, while original location measurements are shown by blue pinpoint and blue circle.

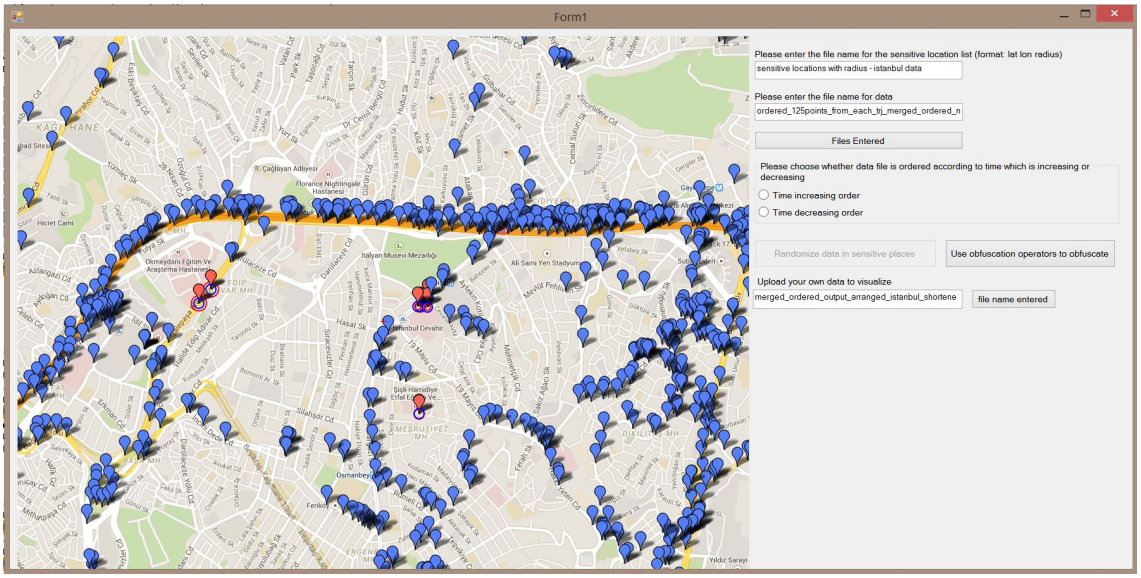


Figure 7.5: Trajectories visualized after obfuscation method of Chapter 5

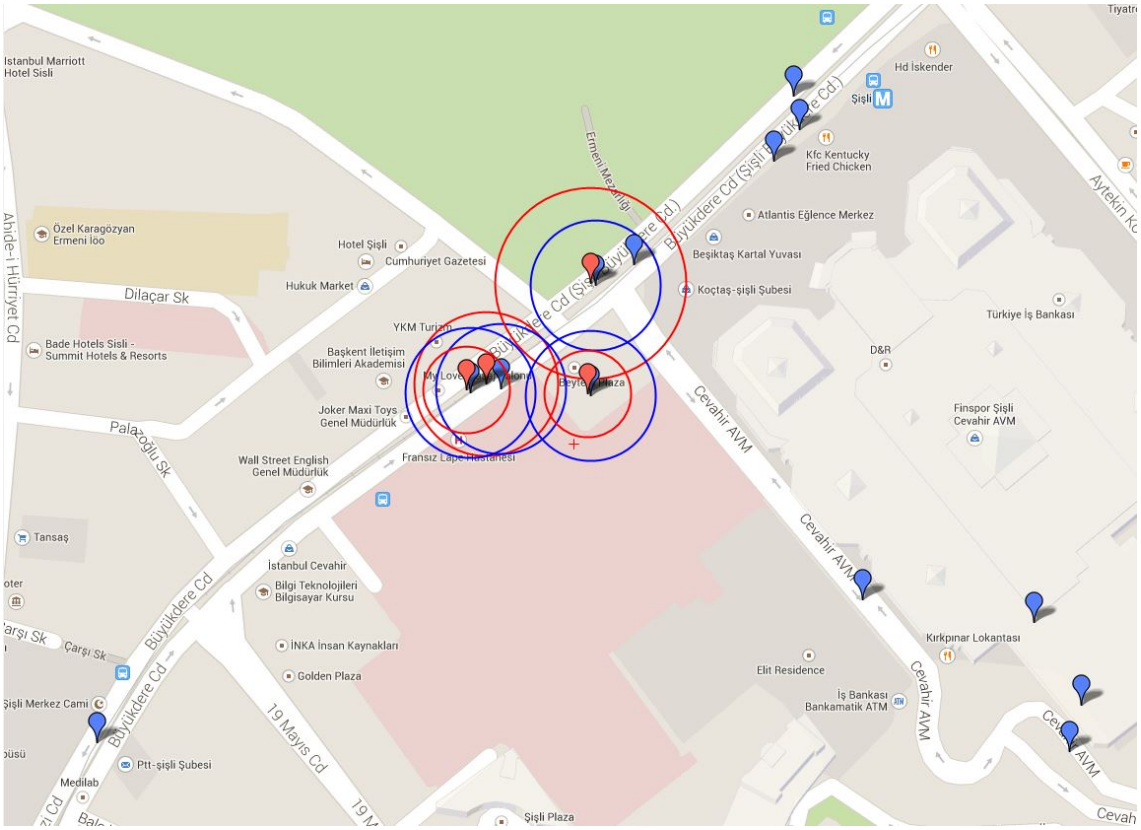


Figure 7.6: Trajectories visualized after obfuscation method of Chapter 5 in detail

Another scenario in which this tool can be used is when a user wants to visualize his/her data. User enters the name of the file and the data is shown. Each data point is indicated with red pinpoint. When the user moves the mouse on a pinpoint, he/she can access the trajectory id and the order of the point in the trajectory. The user can zoom in further to investigate the data. Resulting visualization can be seen on Figure 7.7.

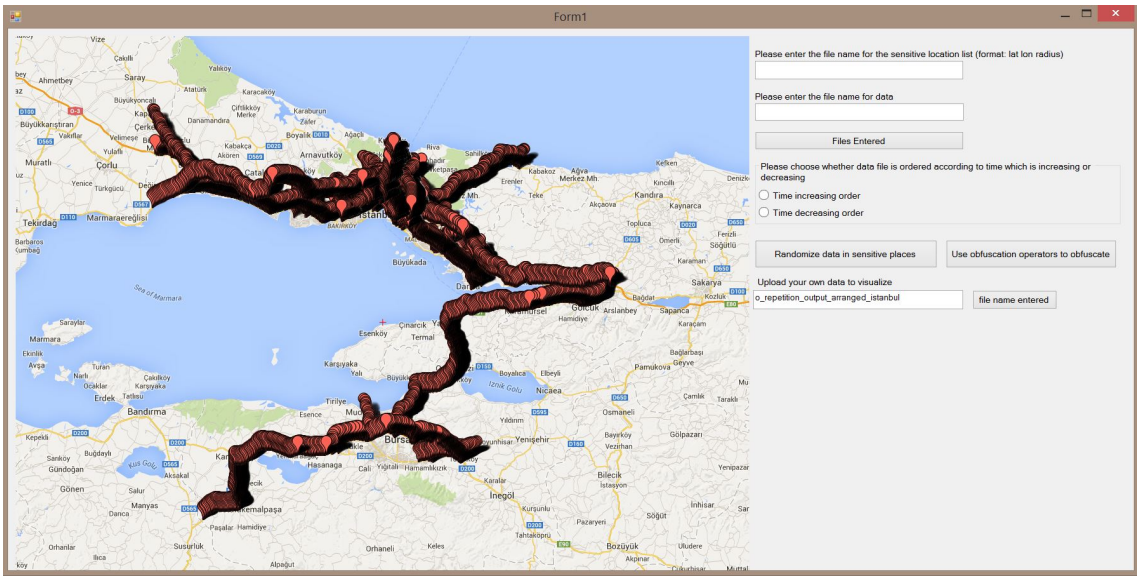


Figure 7.7: Trajectories visualized

7.4 Setting

We have used the tool, mentioned in the previous section for obfuscation methods explained in Chapter 4 and Chapter 5.

7.4.1 Parameters

Parameters used to implement those obfuscation methods are given as follows:

Our obfuscation method needs three parameters, which are average neighboring distance, GPS error in meters, and k anonymity number.

Average neighboring distance is calculated by Equation 4.2. It is simply summing up distances between a trajectory's data points, and it is repeated for all trajectories, then it is divided by the number of data points in the whole data. In this way, average neighboring distance between data points is calculated. For our data, it is calculated as 90 meters.

GPS error is given by the data provider. The GPS error value for our data is 30 meters.

n randomization number is used on the algorithm explained in Chapter 4 on step 4. It is used to further prevent adversaries by including randomization. n value we used in our experiments is 2.

The other obfuscation method mentioned in Chapter 5, uses 3 parameters, r_o, r_i and R_f.

r_o is defined as the GPS error that could be achieved in the perfect environmental conditions. This value is taken from the data provider and it is 10 meters.

r_i is the GPS error in current location measurement which is 30 meters.

R_f is the final relevance to be determined by the user preferences. It is a value between 0 and 1. The important point to note is that it should be smaller than $R_i = r_o^2/r_i^2$. This is essential for the obfuscation to take place. R_i is calculated as 0.1 and the R_f is selected as 0.05.

7.4.2 Obfuscation

Since we are given the sensitive location data, data points falling into those places are our concern. Therefore, we only obfuscated sensitive data points.

We explained obfuscation operators in Chapter 5. For each sensitive point, an operator is selected randomly and obfuscated accordingly.

7.4.3 Confidence of Area

Confidence of area calculation is done as described in Section 3.1.2. Our objective of obfuscation is to hide the information of a trajectory passing by a sensitive location. Therefore, we followed the following approach.

We traversed the original data set, when we came across a data point falling into sensitive area, we calculated the confidence of the sensitive area for the corresponding point in the candidate trajectory set. Thus, the value of the calculated confidence of area is interpreted as can the attacker anticipate that the target trajectory passing by a sensitive place. We used three different candidate trajectory set for our data, which are produced by the attack described in Chapter 6. We calculated confidence of area for each of them.

One of the candidate trajectory set is produced when the original trajectories are used as input for the attack. The other one is the outcome of the attack as a result of the obfuscated input by using the obfuscation algorithm mentioned on Chapter 4.

The third one is the resulting candidate trajectory set when the attack is fed through the trajectories, which are obfuscated with the help of obfuscation operators described in Chapter 5. While calculating the confidence of area, we took into account the radius value produced by those operators for the point which originally is sensitive. When the point is at most (sensitive area radius) + r_f distance away from the sensitive area center, then it is treated as sensitive. While for other candidate sets, GPS error value is used, instead of the r_f.

In our experiments, Confidence of area is calculated according to the same center of the area but with various radius values to see its effects on the attack. When we change the number of known trajectories, attack produces different set of candidate trajectories, so confidence of area calculations are done again for these set of data.

7.5 Results

Results are expressed as confidence of area; the term is explained in Section 3.1.2. In Figure 7.8, Figure 7.9 and Figure 7.10 results can be seen. In figures, k is the number of known trajectories, which is an important parameter of the attack. Graph tagged as "No Obfuscation" is the average of the confidence of area calculated by the candidate trajectories generated when the attack is applied to the non-obfuscated trajectories. "Obfuscated (Samarati)" is the average of the confidence of area in which candidate trajectories are obtained by the obfuscated trajectories, obfuscation method used is explained on Chapter 5. "Obfuscated" is the average of the confidence of area which is obtained by candidate trajectories produced as a result of attacking the obfuscated trajectories, and the method explained in Chapter 4 is the applied the obfuscation method.

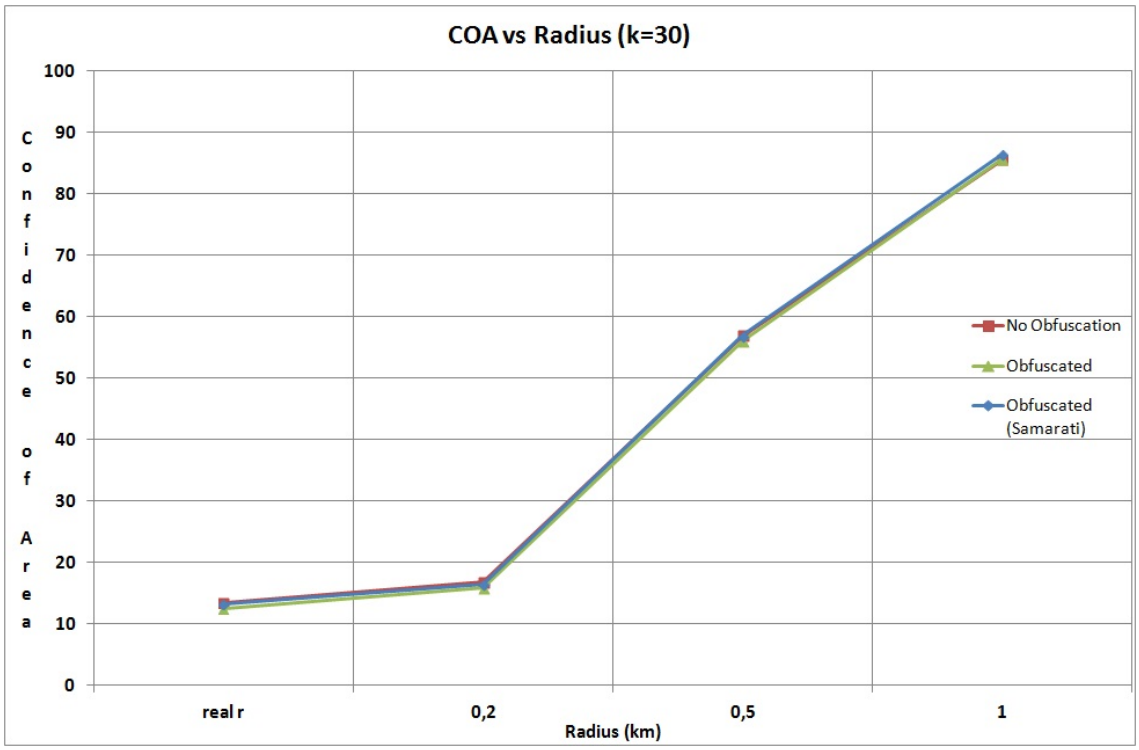


Figure 7.8: Confidence of Area (COA) when k=30, according to the radius of the area in which confidence is calculated, real r is the actual measured radius of the sensitive place

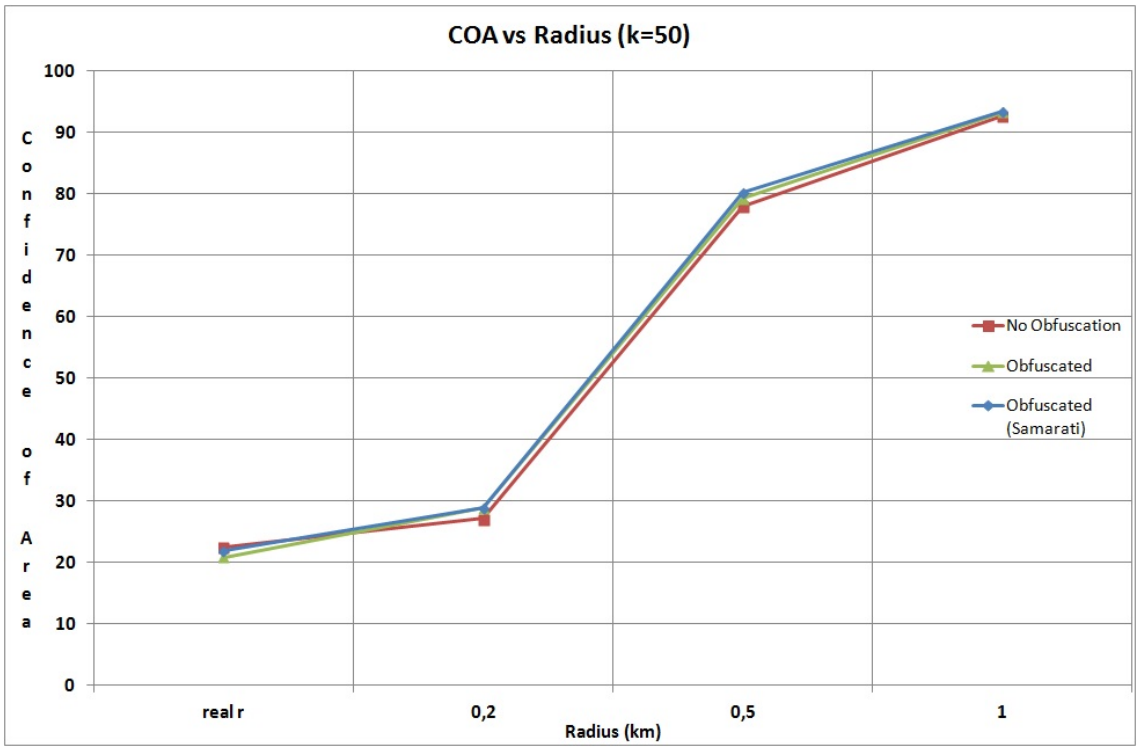


Figure 7.9: Confidence of Area (COA) when k=50, according to the radius of the area in which confidence is calculated, real r is the actual measured radius of the sensitive place

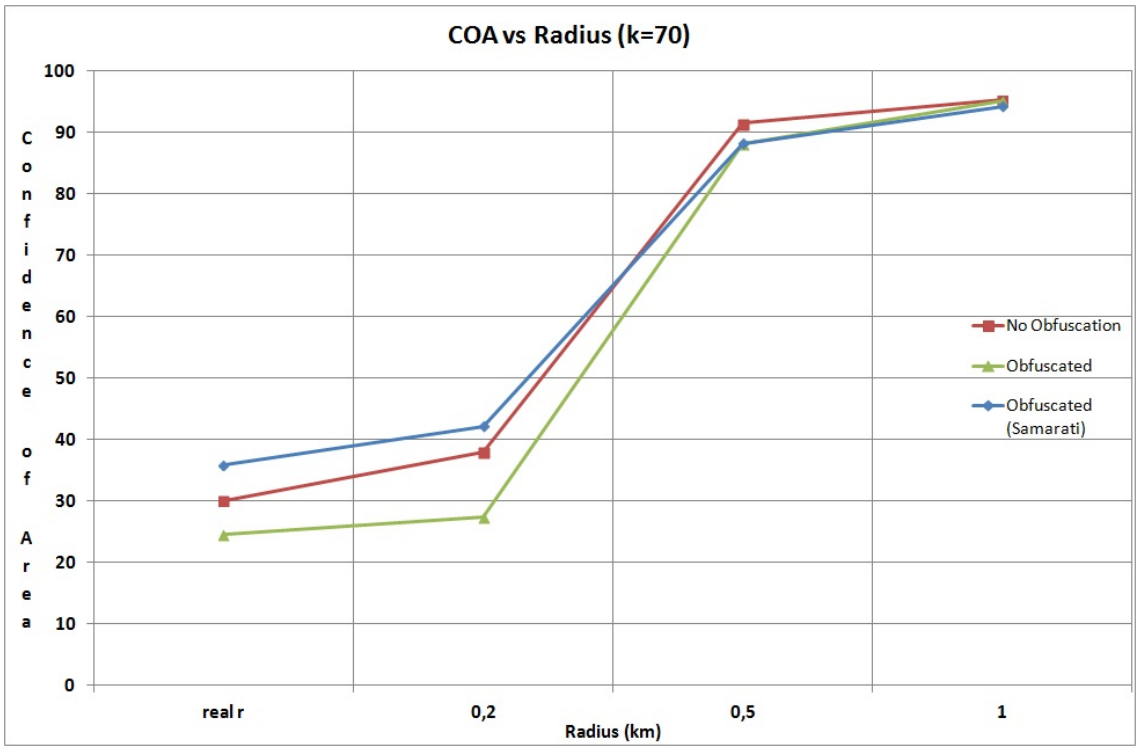


Figure 7.10: Confidence of Area (COA) when k=70, according to the radius of the area in which confidence is calculated, real r is the actual measured radius of the sensitive place

In total 50 original trajectory points are found to be sensitive. The results in Figure 7.8, Figure 7.9 and Figure 7.10 are the average of confidence of areas for each sensitive place, into which a location point falls.

In our experiments, we wanted to see how obfuscation techniques play a role for the attack of [2]. We further investigated their relation to known number of trajectories and the radius of the area of which confidence is calculated. We used k (number of known trajectories) as 30, 50 and 70. Furthermore, we used radius values of the measured real length of the area, which is 44 meters on average, in addition, we used 200, 500 and 1000 meters as the radius of the area while calculating the confidence. For each experiment, results of average confidence of areas are compared. We can see that applied obfuscation techniques are not enough to protect the sensitive data as the attacker calculates the probability distribution of the candidate trajectory high enough to conclude the target resides in the sensitive area. Adversary can obtain the success in an area of 1000 meters in the case of the k=30, while radius of 500 meters is enough for the case of k=50 and k=70. When

we use small radius values, for example, the real r or 200 meters, average confidence of areas are low for all the cases, not particularly for the confidence of areas with obfuscated trajectories. Hence, we cannot conclude that obfuscation techniques are successful for those areas. Furthermore, we had an experiment to see whether for big radius values of 500, 1000 meters, the confidence of areas are always high even though trajectory is not passing nearby. We chose an area in Istanbul, where trajectories were not passing nearby and all confidence of areas was zero, so we were able to verify our results.

Besides, by looking at the results, we are able to conclude that as the number of known trajectories gets higher, adversary is able to find out presence of the target in areas with smaller radius values. Similarly, higher the radius of the area, higher the average confidence of area gets.

Chapter 8

Conclusions and Future Work

In this thesis, we implemented two obfuscation techniques. One of them is a method that we designed. The other one is a state of the art method explained in [1]. Furthermore, we implemented a geo-spatial visualization tool which makes it easier for an end user to specify sensitive locations. The visualization tool uses the obfuscation methods and displays the obfuscated trajectories. Visual validation of the obfuscation methods is achieved through the tool and then, we investigated effects of the two obfuscation techniques on the attack algorithm proposed in [2].

We have devised an obfuscation method explained in Chapter 4. Given sensitive locations, obfuscation is done such that the direction of the movement is considered and the obfuscated point does not fall into the sensitive place anymore. If a location point is sensitive, candidates are formed around its next trajectory point in a circle which has a radius of average neighboring distance. Those points are mapped to the nearest road segment. Among two points, which have the smallest road distance from the original sensitive point, one is chosen randomly as obfuscated point. Thus we aimed at perturbing the trajectory in a manner to prevent any adversary which may observe obfuscation if the data around the sensitive location is perturbed too much.

Besides the one we proposed, we have also implemented a state of the art obfuscation technique explained in [1], which treats the location points as circular areas because location sensing sensors have a finite precision. Furthermore, it is referring the best possible location measurement, and current privacy is expressed in terms of these two location measurements. The obfuscation is done according to the user preference. This method uses basic obfuscation operators which are Shift, Enlarge

and Reduce. While Shift only changes the center of the area, Enlarge and Reduce change the radius of the location measurement. Furthermore, the combinations of them, when they are used in a sequence, are applied such as Shift and Reduce, Shift and Enlarge. Common characteristics of all these obfuscation operators is that they should end up with a final location measurement providing the requested privacy, and the final area obtained should have overlapping parts with the initial location measurement. Thus, the final area is related to the initial area.

In order to evaluate the implemented obfuscation techniques, we tested the attack algorithm explained in [2] on large scale real data set. The attack method tries to find a target trajectory given a small set of known trajectories and their pairwise distances as a Dissimilarity matrix, including the pairwise distances to the target trajectory. Later the found probability distribution around a chosen area is used to anticipate the presence of an individual in this area.

We used the traffic data located in Istanbul and sensitive locations which are chosen as health related places such as hospitals, polyclinics, medical centers. Data is obfuscated by using two techniques described above. Later, the obfuscated trajectories as well as non-obfuscated (original) trajectories are used as input for the attack. Resulting candidate trajectories are used to calculate the confidence of area for sensitive locations. We only calculated confidence of area for the sensitive location points in the original sensitive data, our aim was to test if the corresponding point in candidate trajectory set still appears as sensitive or not after it is obfuscated. We have used different radius values for sensitive locations, although 500, 1000 meters provide enough accuracy for the adversary to conclude that a trajectory is present. When we examined the confidence of areas, it is seen that the obfuscation methods are not adequate for this attack. Attacker can obtain high confidence of area when more than 500 meters are used as the radius of the area to calculate the confidence. It is because the obfuscation methods used do not perturb the data such that the obfuscated point is at least 500 meters away from the sensitive location center. Furthermore, the linearity of the trajectory is not destroyed by those techniques, which further assists the attack.

In addition, it is observed that higher the number of known points for the attack, higher the confidence of area for the sensitive locations. Similarly, as radius of the

area increases, confidence of area increases as well.

As a future work, the obfuscation method of Chapter 4 can be changed such that, when candidates around the next non-sensitive points are snapped to the nearest road, the distance from the sensitive location center could be taken into account. If the distance is smaller than 500 meters, another circle around next non-sensitive point can be formed with longer radius and the procedure can be repeated until such candidates are found. Thus, attacker may not find the trajectory passing around the sensitive place when 500 meters are used as the radius of that place.

Moreover, the technique explained in Chapter 5 may produce a better protection for the attack if the underlying sensing technology has a low precision of 500 meters or more. In this way, it can produce output where obfuscated points can be 500 meters away when Shift operator is included in the obfuscation, though it is not guaranteed.

Bibliography

[1] C. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 1, pp. 13–27, 2010.

[2] E. Kaplan, M. E. Nergiz, and Y. Saygin, "Discovering the whereabouts of private trajectories."

[3] G. Dini and P. Perazzo, "Uniform obfuscation for location privacy," in *Proceedings of the 26th Annual IFIP WG 11.3 Conference on Data and Applications Security and Privacy*, ser. DBSec'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 90–105. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-31540-4_7

[4] "Roger clarke's 'privacy introduction and definitions'," http://www.rogerclarke.com/DV/Intro.html#Aff.

[5] C. Clifton, D. Mulligan, and R. Ramakrishnan, "Data mining and privacy: An overview," in *Privacy and Technologies of Identity*, K. Strandburg and D. Raicu, Eds. Springer US, 2006, pp. 191–208. [Online]. Available: http://dx.doi.org/10.1007/0-387-28222-X_11

[6] J. Domingo-Ferrer and V. Torra, "Privacy in data mining," *Data Mining and Knowledge Discovery*, vol. 11, no. 2, pp. 117–119, 2005. [Online]. Available: http://dx.doi.org/10.1007/s10618-005-0009-3

[7] M. Kantarcioğlu, J. Jin, and C. Clifton, "When do data mining results violate privacy?" in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04.

New York, NY, USA: ACM, 2004, pp. 599–604. [Online]. Available: http://doi.acm.org/10.1145/1014052.1014126

[8] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '00. New York, NY, USA: ACM, 2000, pp. 439–450. [Online]. Available: http://doi.acm.org/10.1145/342009.335438

[9] Y. Saygin, V. S. Verykios, and C. Clifton, "Using unknowns to prevent discovery of association rules," *SIGMOD Rec.*, vol. 30, no. 4, pp. 45–54, Dec. 2001. [Online]. Available: http://doi.acm.org/10.1145/604264.604271

[10] A. Inan and Y. Saygin, "Ubiquitous knowledge discovery," M. May and L. Saitta, Eds. Berlin, Heidelberg: Springer-Verlag, 2010, ch. Privacy Preserving Spatio-temporal Clustering on Horizontally Partitioned Data, pp. 187–198. [Online]. Available: http://dl.acm.org/citation.cfm?id=1980502.1980515

[11] C. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati, "Location privacy protection through obfuscation-based techniques," in *Data and Applications Security XXI*, ser. Lecture Notes in Computer Science, S. Barker and G.-J. Ahn, Eds. Springer Berlin Heidelberg, 2007, vol. 4602, pp. 47–60. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-73538-0_4

[12] C. Bettini, S. Mascetti, X. Wang, and S. Jajodia, "Anonymity in location-based services: Towards a general framework," in *Mobile Data Management, 2007 International Conference on*, May 2007, pp. 69–76.

[13] C. Bettini, X. S. Wang, and S. Jajodia, "Protecting privacy against location-based personal identification," in *Proceedings of the Second VDLB International Conference on Secure Data Management*, ser. SDM'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 185–199. [Online]. Available: http://dx.doi.org/10.1007/11552338_13

[14] M. Mokbel, C.-Y. Chow, and W. Aref, "The new casper: A privacy-aware location-based database server," in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, April 2007, pp. 1499–1500.

[15] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information (abstract)," in *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, ser. PODS '98. New York, NY, USA: ACM, 1998, pp. 188–. [Online]. Available: http://doi.acm.org/10.1145/275487.275508

[16] O. Abul, F. Bonchi, and M. Nanni, "Never walk alone: Uncertainty for anonymity in moving objects databases," in *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ser. ICDE '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 376–385. [Online]. Available: http://dx.doi.org/10.1109/ICDE.2008.4497446

[17] M. E. Nergiz, M. Atzori, and Y. Saygin, "Towards trajectory anonymization: A generalization-based approach," in *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, ser. SPRINGL '08. New York, NY, USA: ACM, 2008, pp. 52–61. [Online]. Available: http://doi.acm.org/10.1145/1503402.1503413

[18] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, ser. ICDCS '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 620–629. [Online]. Available: http://dx.doi.org/10.1109/ICDCS.2005.48

[19] ——, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 1, pp. 1–18, Jan 2008.

[20] M. Terrovitis and N. Mamoulis, "Privacy preservation in the publication of trajectories," in *Proceedings of the The Ninth International Conference on Mobile Data Management*, ser. MDM '08. Washington, DC, USA:

IEEE Computer Society, 2008, pp. 65–72. [Online]. Available: http://dx.doi.org/10.1109/MDM.2008.29

[21] T. T. B. Le and T. K. Dang, "Semantic-aware obfuscation for location privacy at database level," in *Proceedings of the 2013 International Conference on Information and Communication Technology*, ser. ICT-EurAsia'13. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 111–120. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36818-9_12

[22] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, Nov 2003, pp. 99–106.