IT 11 040
Examensarbete 30 hp
September 2011

Protecting the privacy of users querying Location-based Services

Volkan Cambazoglu

Institutionen för informationsteknologi
Department of Information Technology

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Protecting the privacy of users querying Location-based Services

Volkan Cambazoglu

Location-based services (LBS) is a new and developing technology for mobile users. Nowadays, it is very easy for a person to learn his/her location with the help of a GPS enabled device. When this location is provided to a LBS via querying, it is possible to learn location dependent information, such as locations of friends or places, weather or traffic conditions around the location, etc.
As LBS is a developing technology, users might not be aware of the risks that it poses. There have been many protocol proposals aiming at protecting the location privacy of the users, who communicate with a LBS. K-Anonymity is one of the popular solutions that aims to gather k users under a cloak in order to make queries of each user indistinguishable in the eye of an adversary. However, there are claims that K-Anonymity does not solve the problem of location privacy.
In this master thesis, the aim is first to scrutinize existing protocols on location privacy, in order to study their approaches to the problem, strengths and weaknesses. The thesis continues with implementation of an existing protocol and detailed analysis of essential components of the location privacy problem. The thesis is concluded by confirming the ideas on K-Anonymity.

Handledare: Ioana Rodhe/Christian Rohner
Ämnesgranskare: Björn Victor
Examinator: Anders Jansson
IT 11 040
Tryckt av: Reprocentralen ITC

Acknowledgements

I would like to thank Ioana Rodhe, Christian Rohner and Björn Victor for their guidance, support and friendliness during the master thesis. It was a great opportunity for me to learn from their experiences, constructive critics and way of working together. I always felt their confidence in me and worked even harder to deserve it continuously. I am very happy that I have worked with great people in this project, which I consider as my biggest and most important work so far.

I also would like to thank everyone in the Communications Research group for creating such happy and comfortable working environment. It is a pleasure to be a part of the group and welcomed in almost every activity in it.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to Location-Based Services

New types of smart mobile devices enabled the emergence of Location-Based Services (LBS). A user of the service carries a mobile device that obtains its location via Global Positioning System (GPS) [3] or a Wireless Local Area Network (WLAN) [10]. With the help of a service provider, the device can, for example, discover nearby restaurants or whereabouts of a friend [1, 5]. In other words, a user provides a user name, location information in x and y coordinates, a time stamp and a message to the Location Service Provider (LSP). Message content can include a question or a keyword so that the LSP can define where the target is. When the LSP calculates the user's and the target's locations, it returns a result, which might indicate a path from the user's location to the target's location or simply present two locations on the map, to the user.

The advantage of this system is letting users find useful information according to their location information. There are many possibilities of interpreting and using location information. It is not required that there are always two end points that a user starts from and ends at. A user could also retrieve local information, such as weather or traffic conditions, according to the location. Location information could also be used to track a vehicle. A user waiting at a location could follow a vehicle, e. g. a bus, on a mobile device, so that the arrival time of the vehicle or an intersection point on the vehicle's route could be learned.

While the LBS helps users reach places or people easily, private informa-

tion of users could be disclosed to other people. As users do not want their locations and mobility patterns to be revealed to other ones, the aim is to prevent people from making an identity-location binding. Identity-location binding means that one is able to tell that a specific user has been to a specific location.

As LBS is a new and interesting opportunity for users, users might not be aware of the risks that it poses. LBS providing companies/organizations might set policies to protect user's rights. However they might not work all the time or cover all possibilities. As Karim mentions in his paper The Privacy Implications of Personal Locators: Why You Should Think Twice Before Voluntarily Availing Yourself to GPS Monitoring, "There must be significant safeguards to protect the personal, marketable data that a personal tracking device generates from circulation to interested third parties.". [29] It is necessary to pay attention to protecting personal data because of the unclear guidelines for companies when to ask for approval of customer in order to release his/her private data to another entity. In addition to third parties, law enforcement can also by-pass company policies. According to Karim, "The personal tracking device creates a new realm of potential for government surveillance. Law enforcement could intercept an individuals GPS data, or access past information, making the individual constantly vulnerable to surveillance.". [29]

The problem of providing location privacy to users is very wide that there are many aspects to consider, such as different layers and architectures of communication. For example, some researchers approach the problem from physical layer, some from network layer. This project aims to deal with the problem on application layer. The adversary can be defined as the LSP or someone, who has access to the LBS data. Furthermore, there are different architectures for providing location privacy to users. For example, it can be a central architecture in which all the users communicate with the LBS via a trusted server. This solution has strengths such as being easy to implement and maintain, and weaknesses such as forming a bottleneck for performance and becoming a single point of failure. There are also distributed solutions. [45] This project aims to adopt a central architecture for simplicity and working on server side, the LSP, thoroughly.

According to our aims, the studied system, Figure 1.1, is composed of users, a Trusted Server (TS) and a Location Based Service (LBS). Users have mobile devices with which they access the LBS through the TS. The service is based on the location derived on the mobile phone. In addition

to handling communication between the users and the LBS, the TS is the central component that provides location privacy to the users. After getting queries of users from the TS, the LBS prepares answers for queries and sends them to the TS, which hands them to corresponding users.

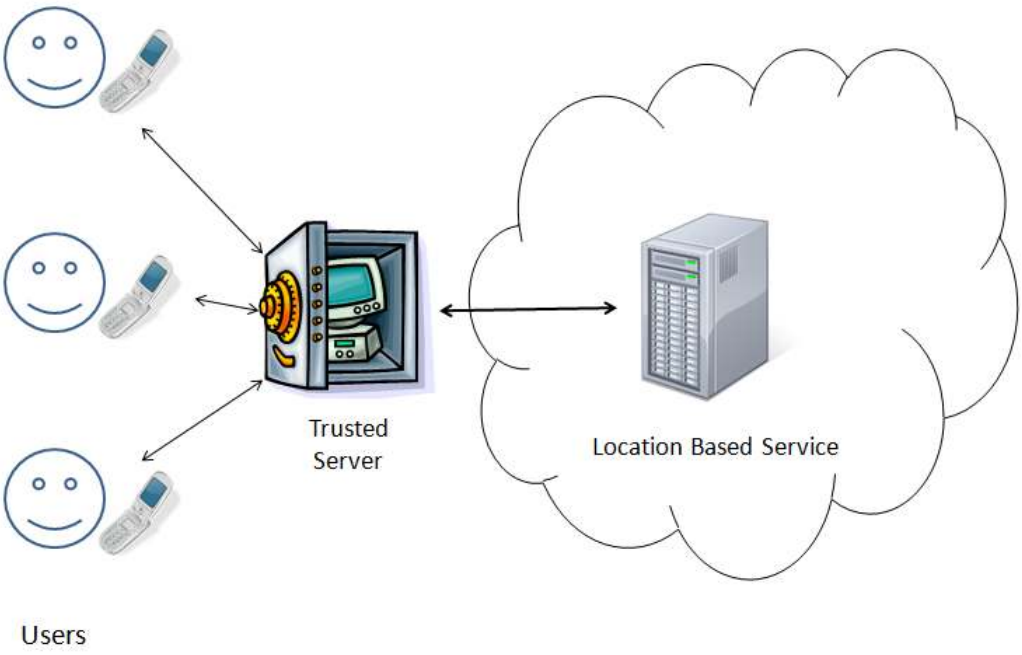


Figure 1.1: System Overview

1.2 The Analysis of the Location Privacy

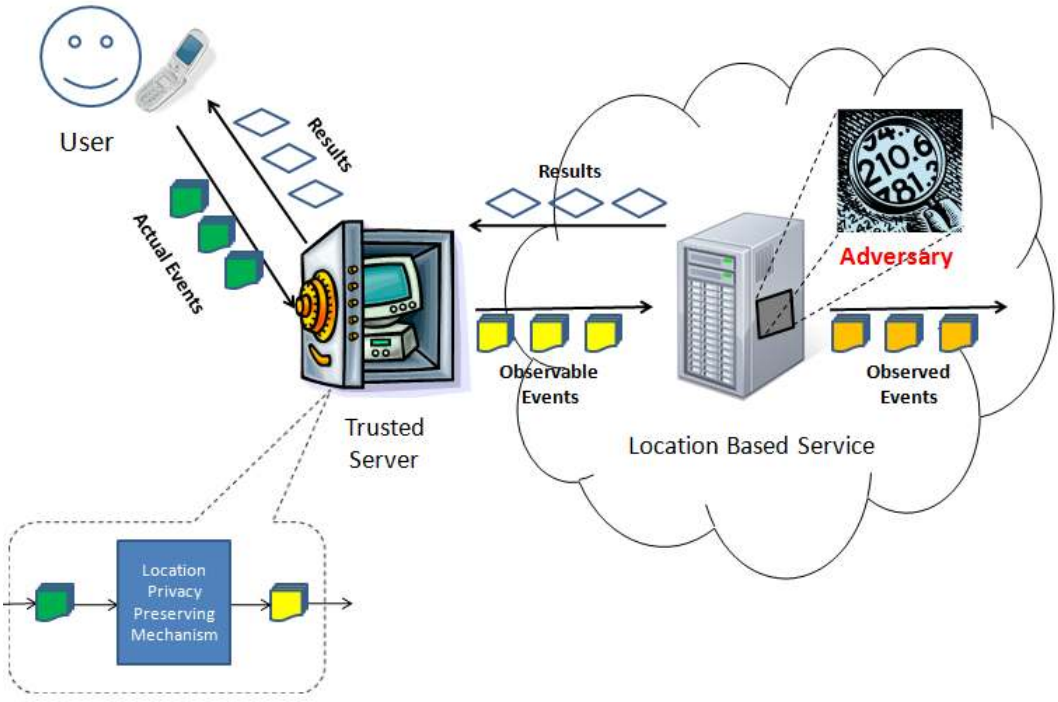


Figure 1.2: Communication between a user and a Location Based Service

Figure 1.2 shows the model to analyze the location privacy of a user assuming an adversary. We will model the capabilities of the adversary in the next section. Users check-in or query the system at locations where they are present. This operation could be interpreted as a generation of an actual event. An **event** [38] is composed of user's identity, location information, time stamp and, optionally, message content. Actual, observable and observed events [38] are all events with different states. State of an event can change due to transformation or observation. Users send their actual events to the TS. The TS modifies the actual events by applying the Location Privacy Preserving Mechanism [38] (LPPM) on them. The LPPM will briefly be explained below, after the explanation of Figure 1.2. When the LPPM is applied on actual events, they become protected or, in other words, location private. Since the resulting events are protected, they can be open

to observation; hence, they are called observable events. The TS sends the observable events to the LBS, where an adversary is present. The adversary observes the events that the LBS has received. The events that the adversary has acquired are called observed events since they are obtained as a result of observation on observable events. The difference between observable and observed events is that the observed ones are a subset of the observable ones.

1.2.1 Location Privacy Preserving Mechanism (LPPM)

The LPPM have been proposed to protect the location privacy of users. When the LPPM is applied on the LBS data, one should not be able to figure out a user's location at a certain time, even if the LBS data and extra information about the user are available. The LPPM can include anonymization, obfuscation, elimination, introduction of dummy events or a combination of them.

- **Anonymization** [12, 20, 38] is applied on actual events so that it is not possible to deduce user's identity by looking at a query or a response. For example, an anonymized query might consist of pseudo name, location information, time stamp and message content. Pseudo name could be anything, such as a random number or name, except the user name.

- **Obfuscation** [11, 14, 21, 22] is a method to make a user's location information and/or time stamp inaccurate or imprecise so that the adversary cannot pinpoint where a user is exactly located.

- **Elimination** [25, 26, 27, 28] means removal of some of the actual events of a user. The reasons might be overuse of the system by the user or privacy degrading parts in the actual event. If a user uses the system frequently and for long periods of time, then that user might reveal too much information about him/herself unintentionally. Furthermore, if a user is staying at a location continuously or always asking for the same content, then an adversary might distinguish the user from others easily. Therefore, it might be necessary to eliminate some of the actual events of users in order to increase their location privacy.

- **Introduction of dummy events** [15, 30, 33, 44] aims to add fake events, which mislead an adversary so that, a user appears to be at a location, which he/she does not really.

Moreover, several LPPM could be combined to provide higher location privacy to users.

- **K-anonymity** [13, 20, 21, 22, 41, 42, 43, 45] is a location privacy solution, which includes both anonymization and obfuscation. Anonymization is applied to protect the user name of the user and obfuscation techniques are applied to protect the location-time couple where the user is present. When an adversary observes the results of K-anonymity mechanism, he/she notices that there are k many indistinguishable events all identity-less and occurring at the same location/area and time period. K-Anonymity will be examined and explained in detail in sections 4, 5 and 8.

1.3 The Evaluation Model for the Location Privacy

As the LPPM is mentioned in the previous subsection, there are various ways of providing location privacy to the users of the LBS. The next step, after applying the LPPM on actual events, is to evaluate the effectiveness of the LPPM. It is also important to benchmark different LPPMs according to their efficiency in protecting the location privacy of the users. It is necessary to understand the adversary model and the location privacy metric, in order to be able to assess the efficiency of the LPPM. Therefore, we, first, look at what the adversary does when he/she acquires the observed events, which are transformation of actual events due to the application of the LPPM on them. Then, we consider a way of evaluating the LPPM depending on the comparison between the actual and the observed traces. The evaluation model is adopted from the paper A Distortion-Based Metric for Location Privacy [38].

1.3.1 The Adversary Model

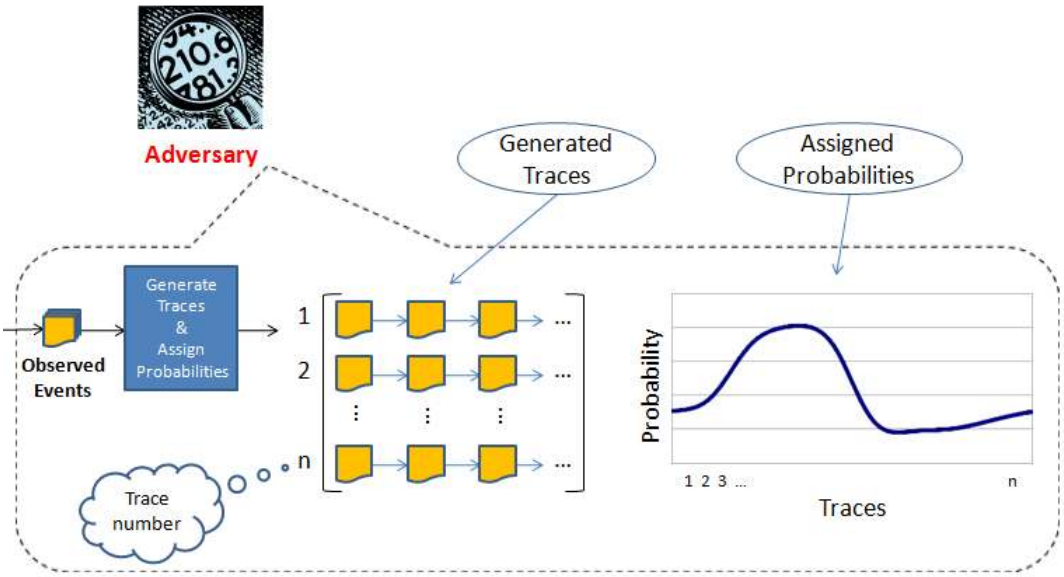


Figure 1.3: Adversary's way of utilizing the observed events

The order of adversary's actions are presented in Figure 1.3 from left to right. The adversary acquires observed events and, then, analyzes them according to his/her knowledge of the users and/or the locations. The analysis consists of generation of traces and assignment of probabilities to them. The adversary generates possible traces out of the set of observed events. The adversary's probability assignment is done according to the order of traces. A **trace** [38] is a sequence of events, which are placed in it in the order they are generated. Each user has an actual trace, which is composed of actual events of the user. There are also observed traces, which consist of observed events. The adversary generates observed traces in a probabilistic manner to figure out the real trace of a specific user. The trace, which is assigned the highest probability, is the closest one to the actual trace of the specified user, according to the adversary.

1.3.2 The Location Privacy Metric

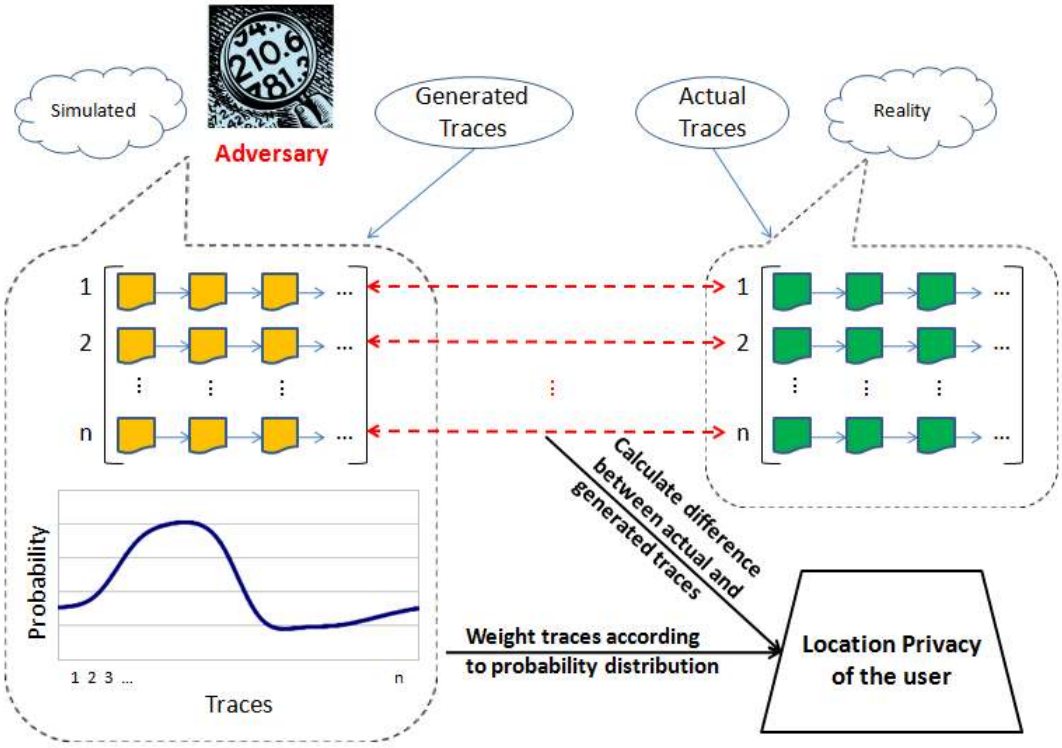


Figure 1.4: Evaluation model for estimating Location Privacy of a user

In reality, the adversary looks at the results of his/her probabilistic analysis and guesses a user's location; however, since we try to simulate the adversary, we think we have what the adversary would have. Therefore, traces, which are generated by the simulated adversary, are compared to actual traces of users. As it is shown in Figure 1.4, the difference between traces, which are weighted according to a probability distribution, tells us adversarys error or how much the observed traces are different or distorted from the actual ones. [38] For example, if the adversary generates a possible trace, which is very close to the original one, and assigns high probability to it, then the adversary's error would be very small. On the other hand, if the adversary assigns low probability to the same trace, then his/her error would be greater than the previous example. Adversary's error is aggregated over

all traces, hence the location privacy of a user depends on complete analysis of the adversary, which means that adversary's both correct and wrong decisions are taken into account. When the location privacy of all users are calculated, it is possible to estimate system wide location privacy.

Chapter 2

Problem Description

There have been many protocol proposals aiming at protecting the location privacy of the users considering approach from application layer and centralized architecture. One of the most popular solutions is K-anonymity, where the locations of k users are cloaked together so that they appear as potential senders of a query. R. Shokri et al. [39] showed recently that constructing cloaking regions based on users' location does not reliably relate to the location privacy of users.

The goal of this master thesis is to investigate the weaknesses of existing protocols, identify the sensitive information revealed, explore how these protocols can be tampered with, and propose a better solution. After completing investigation of existing protocols, there is not enough time to propose a brand new protocol for the location privacy. However, investigation shows that there are still important aspects to consider, which are the traces of users, the adversary model and the comparison of different protocols.

The trace of a user is directly considered in the assessment of the location privacy of the user. It is hard to simulate every possible trace of a user, hence we would like to consider abstract models for traces of users. Generation of traces of users will be explained in chapter 7.

The adversary model is another crucial part of the problem, as it has direct impact on the outcome of the location privacy of a user. Most of the existing protocols on location privacy do not reveal details of their adversary models and we would like see the effect of different adversary models on the location privacy values. Adversary modeling will be explained in chapter 9.

We also would like to be able to compare the existing protocols on the location privacy by implementing them. The reason behind this goal is to

evaluate their strengths and weaknesses; and also to benchmark them. Our investigation also shows that it is hard to compare most of the existing protocols as they do not give the same type of results and do aim to address different aspects of location privacy. For example, both in K-anonymity and Distortion-Based Metric the aim is to provide location privacy to users, however K-anonymity is not suitable for traces and Distortion- Based Metric is suitable. Then, it is necessary to adapt K-anonymity to Distortion-Based Metric, in order to be able to compare them. Moreover, the claims of authors' of Distortion-Based Metric on K-anonymity could be verified by this study. Our simple K-anonymity implementation will be explained in chapter 8, and the results of our simulations will be presented and discussed in chapter 11.

Chapter 3

Methodology

The outline of the methodology that is taken during the master thesis is enumerated below.

1. Look at different approaches to the location privacy problem
2. Confine/narrow the problem space
3. Consider a simple scenario and discuss it
4. Investigate existing metrics
5. Draw an overview of the system
6. Implement the evaluation model
   (a) Generation of traces of users
   (b) Simple K-anonymity
   (c) Probability assignment and the adversary knowledge modeling
   (d) Distance metric

Each step will be explained briefly now.

The project started with identifying important papers on Location Privacy. These papers were selected either as highly cited ones from Google Scholar [7] or among the set of references of papers that are known to be outstanding works. As they will be mentioned in the related work section,

there are different approaches to location privacy such as solutions at different communication levels, which are physical, network and application layers of the communication stack. After examination of these works, it is decided that studying the protocols, which are on the location privacy problem, on application layer and providing a solution to an observed shortcoming would be an interesting and also a feasible project.

The investigation of protocols on application layer is done in more depth in comparison to previous one; because it is necessary to consider all aspects of a certain problem so that a solution could be proposed in a limited period of time. For example, it is figured out that there is a major division of protocols on application layer according to the targeted architecture. Centralized architecture is chosen for the study; because it is simpler to understand, implement and handle in contrast to the distributed one. It is also decided that if there is still time after simulating centralized architecture, we would also look into distributed one. There are also other details, which will be explained in the following sections, such as the adversary model and the evaluation model. All of these details are decided at the second stage of the project.

Since the project started with the aim of investigating K-anonymity protocol, before starting with the implementation, a simple scenario is studied on paper in order to see how the scenario is altered after applying the K-anonymity protocol and if there are any obvious shortcoming of the protocol. This study made us think about the details of the protocol so that it was helpful for the implementation. We also brainstormed about aspects that might pose problems and the weaknesses, which caused those problems. For instance, two of the important aspects from this study were the message content and people that query from the same location. We noticed that message content might reveal identity of a user and when people query the system from the same location, K- anonymity cannot provide adequate protection; because of the absence of the cloaking box.

A detailed analysis of ten major works on location privacy continued even after the study of the simple scenario. The reason of why this process took so long was the complexity of the subject and the limited length of the papers. New aspects and ideas were taken note of after each reading, which shows that the scope of the subject is very wide and each work is only able to cover a part of it. Furthermore, each paper is limited to approximately ten pages, hence the content in each paper is mostly composed of the results and important points. Even if the papers are well motivated and structured, as

the details of each work are invisible or unclear to us, it is generally hard to reflect the solutions, which are on paper, in the code. Significant ideas from these works and how they are included in our implementation will be mentioned in the following sections of the report.

Completing the analysis of papers let us draw the system overview in Figure 1.2, which is presented in the introduction section. System overview, Figure 1.2, helps to prioritize the sequence of our implementation. Each entity and procedure in the system overview are reflected in the implementation as close as possible. Thus we have separate modules and methods to represent the entities and their operations. All of these components interact with the central mechanism as the architecture is chosen in that way. Further details will be mentioned in the implementation section.

During the implementation, we encountered some obstacles about adversary modeling, distance metric and generation of traces. We considered different probability distribution functions in order to model the adversary. We also needed to consider each function for different users, thus the process took time, as a result of implementing, simulating and comparing results with other distributions. For the distance metric, there were some unclear parts and they will be mentioned in section 10. Moreover, generation of traces was another varying and time consuming part. We started with consideration of realistic scenarios that were composed of few events, which did not produce convincing and generalizable results. Then we moved on to automated generation of traces, which required rather simple models that were not necessarily realistic; but having more events included in them. Generation of traces will be explained in section 7.

Chapter 4

Related Work

4.1 The Location Privacy Protocols on Different Layers

This master thesis focuses on application layer protocols on location privacy, however we also looked at various works that approach the problem from different layers of the communication stack. There are some works that consider protection of the location privacy of users by focusing on physical layer. For example, there are use of RF fingerprinting [28], random silent period [36] and MIXes in mobile communication systems [18], in order to protect location information of users from physical layer. Some other works approach from network layer. For instance, pseudonyms, mix zones [12] and anonymous on demand routing [31] are some of the works that aim to achieve the location privacy inside the network. The rest of this chapter is about related work on application layer.

4.2 The Location Privacy Protocols on Application Layer

4.2.1 K-Anonymity

K-anonymity [22, 42, 20, 43, 13, 41, 21, 45] is a popular solution for providing location privacy to users. The concept comes from achieving privacy in data mining, such that when relational data including private data of many users

will be released, K-anonymity protection mechanism is applied on the data to protect privacy of users. K value means that there are k many same values for unique identifiers of users; because if all of the unique identifiers of users look the same, someone cannot link an entry in the data to a specific user. This concept is imported into the location privacy subject. The unique identifiers of a user in location privacy could be considered as user name, location-time couple and in some cases the message content. Therefore, users, who benefit from K-anonymity, are stripped from their user names and cloaked under the same area. An observer would notice that output of K-anonymity solution is composed of identity-less, k-many events all occurring at the same area and time period.

Since one of the aims of this project is to investigate existing protocols on location privacy, the investigation started from K-anonymity. It has both strengths and weaknesses. For example, when a user is located in a crowd, K-anonymity can provide fast and simple solution. Since there are a lot of people around the user, it is very easy to form a cloaked region that users can hide underneath it. If the user is present in that area randomly, he/she can rely on K-anonymity. However, its weakness is the k value and working in a discrete and independent manner. Use of k value comes from a data mining point of view and it is not suitable for preserving location privacy most of the time. For example, an adversary might have knowledge about a user's home and work locations. In that scenario, even if the user is benefiting from k-anonymously cloaked region, which is around location of home or work, he/she is visiting the same location/area over and over again. Therefore the k value loses its effectiveness. In addition to k value, the protection mechanism does not count in history of the user. Therefore, a system cannot guarantee that a user's trace is secure from the beginning to the end, even if the user is cloaked k-anonymously all of the time. One of the papers that lay inefficiency of K-anonymity in protecting location privacy of users is [39].

4.2.2 Metrics for the Location Privacy

Authors of [39] published [38], which is an extensive analysis of existing protocols for the location privacy, later. Apart from K-anonymity, there are uncertainty-based metrics, clustering error based metrics and traceability-based metrics. Shortcomings of existing location privacy mechanisms are explicitly shown in [38].

Uncertainty-Based Metric

Uncertainty-based metric considers only the entropy of events of a user. It is a very general solution. It is not suitable for estimating the probabilistic nature of the adversary. It is very hard to model the adversary; because the adversary's knowledge and probability assignment are unknown. Besides, the adversary can choose wrong events as favorite. Thus, the accuracy of the adversary is another variable in the system. Uncertainty-based metric cannot capture this kind of detail. It is also not suitable for calculating tracking errors that is identification of traces of users. [38]

Clustering Error Based Metric

In clustering error based metric, adversary gets observed events and partitions them into multiple subsets for each user. The error in partitioning indicates the location privacy of the system. Here, the observed events are transformation of the actual events. For instance, a mechanism, such as anonymization or obfuscation, etc., is applied on actual events in order to protect location information of the user from disclosure to public. In this metric, there are two problems that are calculation of set of all possible partitions and suitability for tracing. The set of all possible partitions seem like a very big set, however, it is not. As the observed events are not independent events, some of the events are associated with each other according to location and time. Clustering error based metric cannot measure this aspect. For example, a cluster might include two events with the same time instance, which is not possible in a trace. It might also include two events, with consecutive time instances, that are far apart from each other. It might not be possible for a user to cover that distance in the specified period of time. Finally, this metric cannot indicate a user how much location privacy he/she has at any time or location. [38]

Traceability-Based Metric

Traceability-based metric aims to estimate certainty of an adversary in tracking a user. It is mentioned that a user will be traceable until a point in time or location. This point is called a confusion point; as the adversary's uncertainty is above a threshold. [38] It is also mentioned that querying the LBS periodically, in time space, exposes sensitive locations for users. They suggest that querying the LBS can be done based on areas, which means that

the users do not send queries or the LBS does not expect queries at some locations, which are private areas to use the service. Those places are out of the range of the service. On the other hand, when some places are defined as service points, a user, who passes by that location, has to send a query or the LBS extracts the information from the device of the user. Therefore private locations of users become protected in terms of location privacy and users still benefit from the system in other places as they are traceable. [25]

Distortion-Based Metric

The set of criteria, which is used to evaluate existing location privacy metrics, is composed of adversary's probability of error and tracking error, users' actual traces and private location-time couples, measurement of traceability of users, genericity of the metric and the granularity of the resulting location privacy value. Each criterion reveals more insight about the problem and existing metrics. For example, adversary can make mistakes; but uncertainty based metrics or K-anonymity metric is not able to count in adversary's error in probability assignment or tracking users. Furthermore, considering actual traces of users at all times is also important, because it helps to assess how successful the adversary is in tracking a user. Location/Time sensitivity is another helpful criterion such that private location-time couple(s) of a user could be handled with caution, because if users visit same locations over and over again, then the location privacy mechanisms might struggle to protect users at those locations and times. Being able to measure traceability of users is crucial; because the events that are part of a trace are related to each other. The metrics as K- anonymity or clustering error based do not consider traces of users, instead they only look at individual events, which is a shortcoming for both metrics. The metrics are evaluated also according to their capabilities of measuring impacts of different LPPM that is the genericity of the metric. It is expected that all metrics should be able to capture the effects of different protection mechanisms. The last criterion considers the granularity of the measurement, which means that a metric should be able to indicate a user how much location privacy he/she has at a certain time. If this can be done, then it is also possible to estimate system wide location privacy. [38]

Moreover, authors of [38] provide another solution, which is the Distortion-Based Metric. They claim that their metric satisfies all of the criteria mentioned above. The Distortion-Based Metric aims to reconstruct actual data

by investigating observed events. Reconstruction is done by hypothesizing relationship among observed events and replacing them with possible representative events using probability. In other words, they try to reduce uncertainty and predict actual traces of users. There are, of course, many assumptions in the paper. For example, representative events are computed using adversarys knowledge of users, which is not defined in the paper. Another example is events happen consecutively, which means that there is a defined time gap between two events. Thus if something is missing, then it is known to be eliminated.

Chapter 5

Simple Scenario

After discussing K-anonymity metric, a simple scenario is studied on paper to see how the mechanism works.

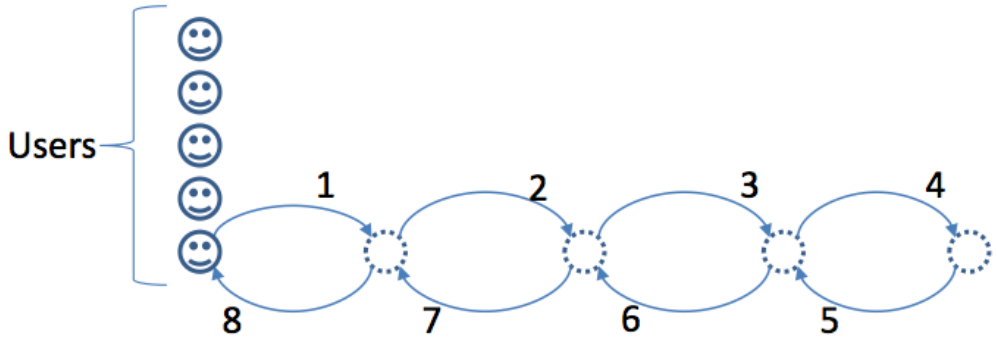


Figure 5.1: Simple Scenario

The scenario shown in Figure 5.1 consists of five users grouped at the same location. One of the users departs away from the other users in a straight line for 4km and returns to the group along the same path. The purpose of this scenario is to illustrate K-anonymity and to study the resulting location privacy for the departing user and the group.

Events are generated at eight time instances (t_i), evenly spread over the duration of the move. During each time interval, the user therefore moves 1km.

In a first step we look at the observed events without using K-anonymity. We assume that the users apply pseudonyms to hide their identity.

Pseudonym	X coordinate [km]	Y coordinate [km]	Time stamp	Content
a	x	y	t_0	c
b	x	y	t_0	c
d	x	y	t_0	c
e	x	y	t_0	c
f	x	y	t_0	c

Table 5.1: Events at time t_0, when K-anonymity is not used

Pseudonym	X coordinate [km]	Y coordinate [km]	Time stamp	Content
g	x	y	t_1	c
h	x	y	t_1	c
i	x	y	t_1	c
j	x	y	t_1	c
k	x + 0.6	y + 0.8	t_1	c

Table 5.2: Events at time t_1, when K-anonymity is not used

Pseudonym	X coordinate [km]	Y coordinate [km]	Time stamp	Content
l	x	y	t_2	c
m	x	y	t_2	c
n	x	y	t_2	c
o	x	y	t_2	c
p	x + 1.2	y + 1.6	t_2	c

Table 5.3: Events at time t_2, when K-anonymity is not used

The other observed events are generated in the same fashion.

Without K-anonymity, an adversary can conclude that one user is moving away from a group of four users staying at the same location. A mapping of the observed events of the moving user to its pseudonyms is possible. However, it is not possible for the adversary to conclude the identity of the moving user.

In the second step we show the observed events resulting from the K-anonymity mechanism where both location and time information is cloaked. The other assumptions are the same as in the previous run.

The observed events might look like as in the tables below. t_t is the toleration value in time. Let us assume that all users tolerate enough in x and y coordinates, in order to cover everyone in a single cloak, which has k value equals to 5. As it is visible in the table for the first time instance, all of the users seem to be located on a point; because toleration values are only used to calculate if a cloak of k users could be formed. The observed events are cloaked under the minimal and imaginary box that covers all of the users. If all of the users are at the same location, then there is no minimal box, instead it is a point.

Pseudonym	X coordinate [km]	Y coordinate [km]	Time stamp	Content
a	x	y	[t_0 - t_t, t_0 + t_t]	c
b	x	y	[t_0 - t_t, t_0 + t_t]	c
d	x	y	[t_0 - t_t, t_0 + t_t]	c
e	x	y	[t_0 - t_t, t_0 + t_t]	c
f	x	y	[t_0 - t_t, t_0 + t_t]	c

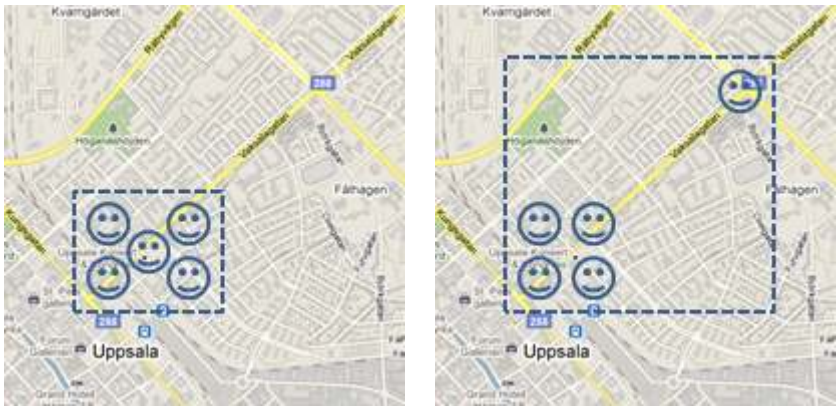
Table 5.4: Events at time t_0, when K-anonymity is used

Pseudonym	X coordinate [km]	Y coordinate [km]	Time stamp	Content
g	[x, x + 0.6]	[y, y + 0.8]	[t_1 - t_t, t_1 + t_t]	c
h	[x, x + 0.6]	[y, y + 0.8]	[t_1 - t_t, t_1 + t_t]	c
i	[x, x + 0.6]	[y, y + 0.8]	[t_1 - t_t, t_1 + t_t]	c
j	[x, x + 0.6]	[y, y + 0.8]	[t_1 - t_t, t_1 + t_t]	c
k	[x, x + 0.6]	[y, y + 0.8]	[t_1 - t_t, t_1 + t_t]	c

Table 5.5: Events at time t_1, when K-anonymity is used

The other observed events are generated in the same fashion. Figure 5.2

(a) and (b)[1] are, consecutively, illustrations of Table 5.4 and 5.5.



(a) The illustration of Table 5.4 (b) The illustration of Table 5.5

Figure 5.2: Illustration of 5 users at different time instances, based on Table 5.4 and 5.5

We can make the following conclusions from using K-anonymity:

As it is mentioned in subsections 1.2.1 and 4.2.1 before, K-anonymity aims to gather k many users within one cloaked area in order to protect them from an adversary. When applied, all the resulting events look the same as it is presented in Table 5.4 and 5.5. For this example, all of the users benefit from k=5, hence five users must be present within each others' toleration in distance and time. Toleration means that a user can be at most t_x units in x coordinate and t_y units in y coordinate far from another one. Furthermore, the events of two users should have at most t_t units difference between their timestamps. When these two requirements are satisfied, then two users could be placed within the same cloaking area.

According to the definition of K-anonymity, if all of the five users are cloaked as shown in Figure 5.2 (b) and for all time instances, then the adversary cannot know who is querying the system. In this scenario, it means that every user has tolerated at least 2.4 km in x coordinate and 3.2km in y coordinate. Adversary knows only the area and it might be a hard task identifying five users in an area of 7.68 km^2. There might be many possible users in that area. However, for the first and last time instances, if the adversary is monitoring x and y coordinates, he/she might figure out who those five

[1]The map is taken from Google Maps [6] .

users are, also depending on the environment.

If the stationary users do not tolerate enough distance in x and y coordinates, and/or period in time as in the previous paragraph, then the mobile user will be cloaked until the edge of the tolerated x and y coordinates, and/or period in time. Then he/she will either try his/her luck with k=1 or not be able to use the service until finding a suitable cloak. If a user uses the LBS with k=1, it means that he/she is on his/her own. In other words, he/she will be visible to an adversary with his/her location, time stamp and message content, just like absence of K-anonymity.

Moreover, message content might also reveal a user, even if the user is cloaked; because an adversary can follow the user at the edges of cloaked areas. Trying to get an answer, as accurately as possible, from the LBS necessitates sending the message content unprotected to the LBS. A user will have a direction, when asking for the same and/or a specific question for a long period of time. A user can get lost in the middle of cloaked regions. However, the user can be identified when switching from one cloak to another. Still, the adversary needs to monitor the user during a period of time.

Finally, the privacy mechanism does not rely only on the k value, but also on toleration of x, y and t ranges. Picking different values affects the outcome of K-anonymity mechanism, because if the toleration values are too small, it might not be possible to form cloaks of k users. When a user is not present within the tolerated area of other users, then that user is left out of the group, which means that he/she either reveals his/her location-time information to the adversary or cannot use the service in order to protect his/her location privacy at the specified location and time.

Chapter 6

Implementation

So far we have analyzed related works on the location privacy and worked out a simple scenario on paper. We have studied the communication between a user and a LBS in detail, as in Figure 1.2. We have also learned how to evaluate the location privacy a user can experience. As mentioned in chapter 2, the goal of this master thesis is to study existing protocols, which provides location privacy in location based services, and identify their strengths and weaknesses. It is necessary to simulate essential components of the LBS and the LPPM, in order to achieve our goal.

We realize that there are certain parts, which need to be analyzed in detail, in order to be able to understand the location privacy problem better. We separate the key components into two groups, which are varying and non-varying parts, because the analysis can be done easily and clearly when there is less or no variation in the analyzed component. On the other hand, if a component has too much variation in it, it might even be very hard to define the scope and parameters of the component.

The unknown and varying parts are identified as the traces of users and the simulation of adversary. The rest of the components are somewhat known or predictable. After being able to make this distinction, we needed to start implementing the key entities of the system, which were user, event, trace and adversary. These entities were represented in the implementation as explained below. In addition to the key entities of the system, the evaluation model for the location privacy was also identified as the Distortion-Based Metric [38]. When the basic components of the implementation was done, we moved on to the varying parts, which will be explained in detail in the following chapters.

6.1 User

- A user of the LBS has a user name so that the trusted server can identify it. User name is kept in a String variable.
- A user can define location-time sensitivity (LTS) [38], which means that being present at a specific location and time is sensitive for the user. If there is a sensitive location-time couple for a user, then no one should be able to track the user at that time and location. This also means that if the user is at a different location at the specified time or at the specified location at another time, then there is no sensitivity for the user. Sensitivity indicates need for privacy by the user. LTS is kept in a HashMap<String, Float>. For example, "(1.0, 2.0), 18:00" is the String value that represents the location-time couple and 0.0 is the Float value that represents the highest sensitivity. The lowest sensitivity is 1.0.
- A user has a reference number that is to be used while querying the LBS. Reference number is increased by one after each query. The reference number is used to get pseudo name of the user. User name and reference number are concatenated and given to MD5 hash computation, thus a user can get a different pseudo name at each query. There can be more secure ways of obtaining a different pseudo name at each query; however it is good enough for this case. The reference number is kept in an int variable.

6.2 Event

- An event is a quadruple of <user's identity, location information, time stamp, message content (optional)>.
- An event belongs to a user, hence it has a user variable in it. Someone can look at an event and see whom it belongs to. The user variable is of type User, which is explained above.
- The location information of the event is kept in two Float variables that are x and y coordinates. X coordinate corresponds to latitude and y coordinate corresponds to longitude.

- Time of the event is kept in a GregorianCalendar. Date and time of the event can be stored in detail in this calendar object.
- An event has also a message content that is kept in a String variable. This message content tells a LSP how to use the location information of the user. It can be empty so that it only says a user was present at this location and time or it can ask for something depending on the location and/or time.
- As it is mentioned in the introduction chapter, an event can be an actual or observable or observed event, however only the actual and observed events are included in the implementation. Actual and observed events are kept in different Vector<Event> objects in the implementation. Vector can be seen as a LinkedList or a flexible array.

6.3 Trace

- A trace is a sequence of events. The events occur at consecutive time instances and are placed in traces in the order they are generated. For example, if there are three events in a trace, time of the first event is before the time of the second event and time of the second event is before the time of the last event. There are no events with the same time instance in a trace. Consecutive time instances means that there is a certain time gap between each event. For example, in the implementation, the time gap is defined as 30 minutes, hence if the first event of a user is generated at 01.00, then the next event will be generated at 01.30.
- Trace is kept as a LinkedList<Event> in the implementation.

6.4 Adversary

- An adversary gets observed events and extracts information, such as number of users and number of time instances, from them. However it is not necessary that adversary figures these details out from the observed events. The adversary might know the number of users in the system by looking at the events that are acquired beforehand or another way.

- We model the adversary by assigning probabilities to traces. A powerful adversary can identify traces with events from the same user and assign them high probabilities, while a weak adversary cannot distinguish the traces and thus assigns the same probability to every trace. In our study we chose different probability distributions to model different types of adversaries.

- At first, we tried HashMap<Trace,Float> to store probabilities of traces, which are generated exponentially for each user. In other words, there are possible traces that a user can take and it is decided by looking at the number of users and the time instances of observed events. For example, if there are 3 users and 4 time instances, then the number of possible traces are 81 (3^4). Therefore, let us call this procedure the "**exponential generation of traces**". After all the traces are generated, each one of them is mapped to a probability. HashMap is a useful data structure to map from traces to probabilities as we only generate distinct traces, which means that there is no collision among traces while being mapped in the data structure.

- As we have found an alternative to generating all possible traces, the mapping from Trace to Float is replaced with mapping from String to array of Float. An example for String object is "(0,0)", which means that "first user at first time instance". An example for array of Float is [0.4, 0.3, 0.05, 0.25], which means that the first user can take one of four paths from first time instance and the probabilities for taking each path is kept in the array. In this example, the adversary believes that first user probably took first path at first time instance. We can call this procedure the "**no generation of traces**"; because there is no need to build a structure for traces in this approach. The traces are distinguished as a result of decision making of which path the user has taken at each time instance.

- Two implementation choices that are explained here will be compared in detail in section 9.

- As it will be explained in chapter 9, the probability distribution functions that are used for modeling the adversary are Uniform, Binomial and Unit Impulse functions. Binomial and Unit Impulse functions are also shifted at each run in order to simulate adversaries with different

favorite traces, which are very likely traces taken by the users. The values that exist for each function are calculated beforehand and stored in the HashMap for corresponding Trace or String value. For example, if there are four traces and the function is Unit Impulse, then the values are [1.0, 0.0, 0.0, 0.0]. In this case, first trace has 100% probability and the rest of the traces have 0% probability.

6.5 The Central Mechanism

Even if the key entities of the system are defined, there is still a need for a central mechanism that the entities could interact with. At this point, there was an idea of combining different location privacy metrics into one application, because every metric is useful for some scenario and has strengths and weaknesses that is different from the other ones. If several location privacy metrics could be combined in one application using fuzzy logic, it would have been possible to handle various scenarios at one instance. The flow of the design could be explained in these steps:

1. Users' mobile devices obtain location information and send it to the adviser.
2. The adviser assesses the situation by looking at a set of messages from users and sends the results to users.
3. Users choose an option from a set of possibilities that the adviser recommends and send their queries to the trusted server by using the chosen option.
4. The trusted server anonymizes the messages of users and hands them to the LSP.
5. The LSP answers users' queries and sends the answers to the trusted server.
6. The trusted server hands LSP's replies to users.

In this design, the adviser seems as another entity in addition to the trusted server and LSP; but it can also be included in the trusted server. Thus the design of the program is made in a way that a location privacy

adviser, which is combined with the trusted server for simplicity, is placed at the center of the program/system and the entities interact with it.

Since we have investigated several existing solutions before starting the implementation, we were able to select Distortion-Based Metric as a starting model for the central component of our application. The reason behind selection of Distortion-Based Metric was its well defined structure and the detailed analysis of other existing metrics in the paper. We thought that this metric covers more details on location privacy problem and it might be easier to evaluate our design, as Distortion-Based Metric has a sound explanation. As we were using an Object Oriented Language, it should be easy to use a model for testing and then replace it with another module. Here the module is the Distortion-Based Metric.

Furthermore, while implementing Distortion-Based Metric, we encountered several obstacles. The paper proposes a model how to evaluate location privacy. An essential point is the assignment of probability distributions to traces. The paper left it open how to assign these probability distributions. In addition to the probability assignment, the distance metric and the location-time sensitivity of users were also explained on an abstract level in the paper. We looked into these unclear parts in detail by getting queries from imaginary users and simulating possible traces of them. The sequence of operations are ordered as below in the code in order to analyze the unclear parts. The details of the overview, below, will be explained further starting with chapter 7.

1. We manually generated traces of several users (3-4) on the map.

2. We, then, selected locations, which have half an hour of time difference between each of them, on the trace and stored these locations in an input file.

3. Application is started by reading the input file. All of the events that are read are kept as actual events.

4. Location privacy preserving mechanism(s) is/are applied on actual events so that observable events are generated.

5. Adversary acquires observed events and analyzes them as explained in section 1.3.

6. Location privacy of each user and system are calculated by comparing actual events and adversary's analysis.

7. Location privacy values are output to a file to be plotted on a graph.

The reason for selecting period of half an hour is to have fairly logical privacy values; because having short period of time between two events decreases location privacy of the user as he/she is overusing the system. However if we leave long period of time between two events, then the system might become useless from users' perspective or seem unrealistic. Half an hour might be considered as a rough value between two extremes.

The starting idea of combining different location privacy mechanisms in one application using fuzzy logic is abandoned, because of time constraints and complexity of the location privacy problem. While trying to implement Distortion-Based Metric, understanding and implementing how traces are evaluated, how distances are calculated and how the probability distribution(s) are applied took longer than expected. After running simulations on these things and agreeing on what is meant, we continued with a simplified K-anonymity implementation; however, time of the project was up at that stage and our first idea was left as a future work.

6.6 Implementation Tools

- The implementation is completely written in Java [8].
- The development environment is Eclipse IDE [2].
- OpenOffice.org [9] is used for spreadsheets. The results are analyzed and plotted in graphs.
- Gnuplot [4] is used for some of the plots, specially for 3D plots.

Chapter 7

Generation of Traces of Users

Our observations showed that when users have certain movement patterns, plotting of location privacy values on a graph also takes certain shape. For example, when users cross each others road, plotting of their location privacy values also cross each other. In this example, the users were actually far apart from each other and they perpendicularly cross each other. There was notable difference between location privacy values of users at the beginning. At the point of intersection location privacy values of both users got closer to each other and as they moved away from each other the gap between their location privacy values also got wider. Furthermore, when users moved almost parallel with having some distance to each other, their location privacy values also seemed to follow the same behavior.

These observations were derived from four scenarios each having twelve events from three users and also a scenario in which there were sixteen events from four users. All of the traces were generated manually by selecting real coordinates on the map. One of the scenarios was provided in Figure 7.1. We wondered if our observation could be supported with more users and events. Generating more events manually could be a time consuming task, hence we decided to use an abstract model to automatically generate many events in a fast manner. Moreover, we decided to work on three models of traces that cross each other, are parallel to each other and form a closed shape. They are explained in the following sections.

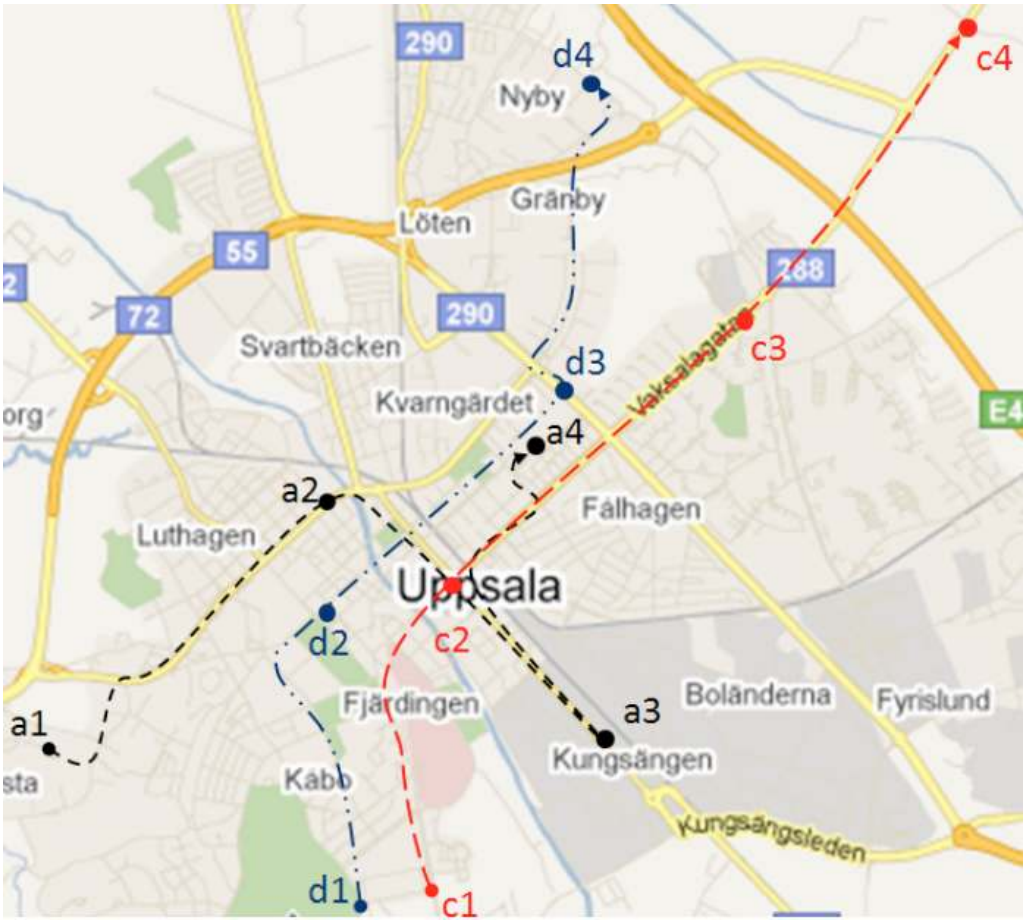


Figure 7.1: Manually generated traces of 3 users, which are user a, c and d, at 4 time instances, which are 1-4

In Figure 7.1, there are 3 users, which are user a, c and d, and 4 time instances. For example, a1 means user a at first time instance. The same applies to other events as well. These traces are built using Google Maps [6] such that time difference between each event is half an hour by walking. In this scenario, the expectation is to observe that user c and d experience similar location privacy and as they cross user a's path, their location privacy values decrease. However, when we use this scenario, the actual and observed events are the same, which means that we do not have any protection mechanism such as K-anonymity at this stage of the project. In this case, as some of the observed events and actual events are the same, the

distance between them is zero; but other events have probability, since the adversary is assumed not to know this detail. Furthermore, we use Binomial distribution and Unit Impulse functions on this scenario and they both produce different results, which match with the expectation partially. After getting these results, we abandon manually generated traces and continue with automatically generated traces; because manually generated traces are very limited and do not give results that can be generalized.

7.1 Cross Traces

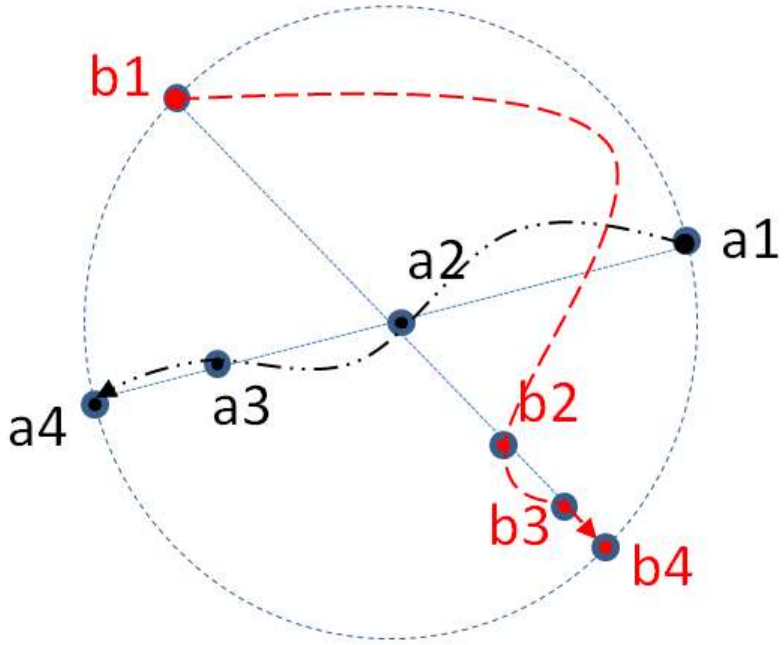


Figure 7.2: Crossing traces. Path in black color belongs to user a and the red one belongs to user b.

Traces that cross each other start from different points, intersect at a point in the middle and then end at different points. For example, in Figure 7.2, there are two users and each one of them has four events at different time instances. (a1 means user a at time instance 1.) A unit circle is drawn and two users are randomly placed on two points that are on the circle. These points (a1 and b1) are starting points of the users and they are distributed between 0 and π degrees. We draw a line that includes the starting point and the origin of the circle, in order to find the end point of the user on the other side of the circle. The events in between starting and end points are, too, randomly placed on the line. Randomness can be obtained over a uniform or normal distribution. This model could be perceived as a zoomed view of an area on a map. A user might be moving within a certain radius of a location and there are also other users that follow the same user behavior. It is easy to model check-in/query locations of users inside a unit circle by

calculating angles and trigonometric values of them. The scaling factor could be considered after preparing the mathematical model. For example, unit circle has a radius of one unit and in reality the radius could be enlarged to whatever distance measure desired and the inflated area could be placed as it is or rotated on the map. Another important thing you might notice that a user's path need not be a straight line. All of the events are generated at consecutive time instances, e.g. each half an hour. A user might cover a long distance by using a vehicle and then a short distance by walking as it could be imagined for user b in Figure 7.2. We rely on the time difference between each event in order to select them on the trace.

7.2 Parallel Traces

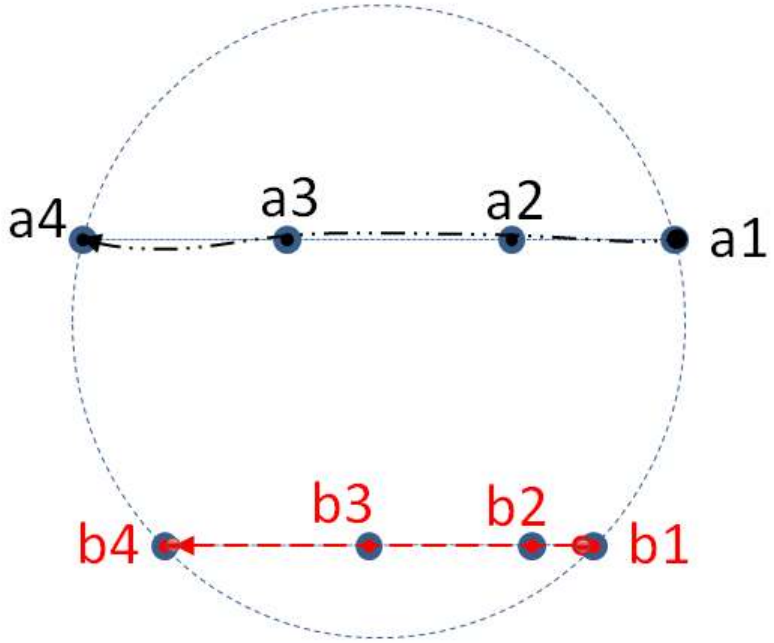


Figure 7.3: Parallel traces. Path in black color belongs to user a and the red one belongs to user b.

Traces that are parallel to each other start and end at different points and they never intersect with each other. For instance, in Figure 7.3, there are, again, two users and each one of them has four events at different time instances. In order to model parallel traces, we, randomly, take a point on the circle between $-\pi/2$ and $\pi/2$ degrees for each user. We subtract the degree, which corresponds to the starting point, from π degrees and calculate the end point. In other words, the starting point is (x,y) and the end point is (-x,y). We again fill the events in between two points randomly, as explained in previous section.

7.3 Circular Traces

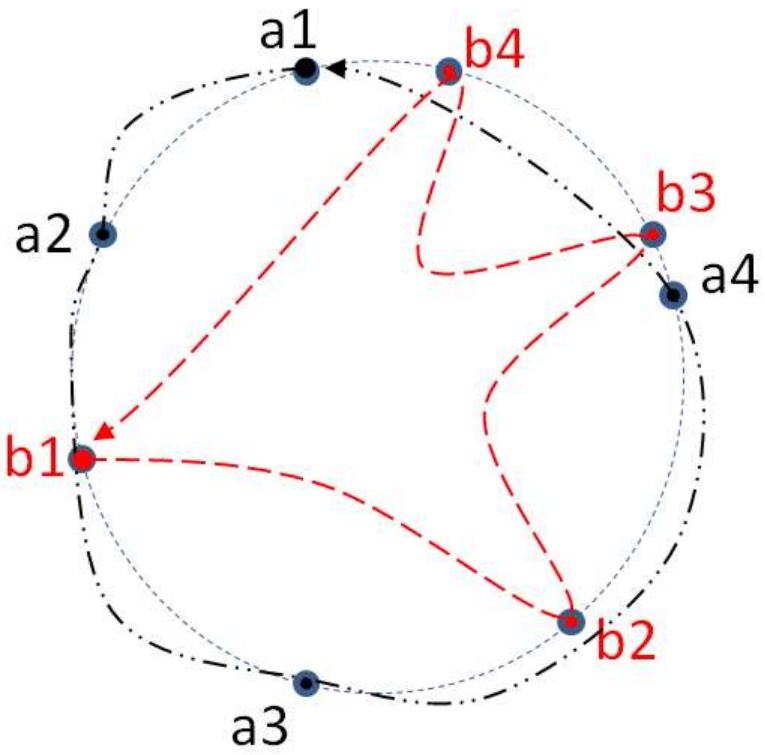


Figure 7.4: Circular traces. Path in black color belongs to user a and the red one belongs to user b.

Circular traces start at a point on the circle, keep moving on it and end at the beginning point. As in the previous examples, we have eight events from two users; but this time they move in a circular shape. The modeling of a circular trace is done by randomly picking a degree from 0 to 2π and then calculating the corresponding point on the circle. A user starts his/her trace from this point and stops at the same point. The points in between them are randomly chosen as a degree, sorted in ascending order and placed on the circle. Again, the actual trace of a user might be different from a circular shape as it is visible for user b in Figure 7.4; however the importance is on placing the actual events on the circle. Moreover, as the number of generated events increase, traces of users are better represented in the model. For

example, in Figure 7.4, user b might follow an irregular shape; because there are only four events of him/her. If there are many events from user b, trace of him/her will take the shape of the circle, as each one of his/her events will be placed on the circle.

Chapter 8

Simple K-Anonymity Implementation

Several approaches to the location privacy are investigated in this project in order to understand different aspects of the problem. However, our focus was mostly on K-anonymity, as we started the project by reading the paper Unraveling an Old Cloak: k-anonymity for Location Privacy [39], which claims that K-anonymity does not address the problem of location privacy in a convincing way. We studied highly cited papers that are on K-anonymity and published in the last ten years. We also examined several works of the authors of [39], in order to see their follow-up works and direction of studies, which lead us to the paper A Distortion-Based Metric for Location Privacy [38]. We adopted the evaluation model for the location privacy from [38] and it is used to assess our simplified K-anonymity implementation.

After thorough examination of existing protocols, we select the paper Location Privacy in Mobile Systems: A Personalized Anonymization Model [20] as an example of K-anonymity solution. As the name suggests, the K-anonymity implementation of this paper considers a personalized approach for each user. As it is mentioned before in the report, K-anonymity aims to cloak k users together so that they seem identical to any observer. Picking the k value is easy when dealing with relational data, however it is hard to figure the value out in real time. There might not be enough users to form a group of k at a certain location or time. If there are not k users around a location, then the system chooses to work with k equals to 1. This is calculated by looking at the toleration values in x and y coordinates, and time space. In this case, where K-anonymity does not work, the user is

clearly visible to any adversary. Another issue while selecting the k value is the performance of the mechanism, because if the mechanism cannot find a solution until the toleration in time space expires, then the events are discarded from the mechanism meaning that they are, again, not protected. In [20], the personalized approach is applied to k and toleration values, so that every user can define their personal values and do not need to meet with system wide values. By this way, users might benefit from the system more frequently, as the scenarios, in which K-anonymity does not work, can be by-passed.

Steps	States	Variables
1	**Actual event**	(user id, x, y, t, C)
2	**K-anonymity**	(k, t_x, t_y, t_t)
3	**Observed event**	(pseudo name, [x-t_x, x+t_x], [y-t_y, y+t_y], [t-t_t, t+t_t], C)

Table 8.1: An example of application of K-anonymity

In the example above, x is x coordinate, y is y coordinate, t is time stamp of event, C is message content, t_x is toleration in x coordinate, t_y is toleration in y coordinate, t_t is toleration in time space and pseudo name can be a random or defined string or number, but it should be different from the user id.

In [20], (k, t_x, t_y, t_t) values are kept as personal values of each user, however we choose to use system wide values for simplicity and automation of simulations. In our simulations, the aim is to protect every user with K-anonymity, thus toleration in x and y coordinates are chosen as the smallest integer value that covers all of the users at once. The actual events are obtained using the automated generation of traces of users, as explained in the previous chapter. Selecting the smallest integer is easy in this case as the actual events are plotted in or on the unit circle, which means that the longest distance between two events could be two units. Therefore, the toleration values in x and y coordinates are both two units. There is another detail related to the toleration values in x and y coordinates that is the location information in the observed event. As users can be present at different locations, when the toleration values in x and y coordinates are applied on them, the resulting areas can intersect; but not cover each other. Thus, the toleration values are only used to check if a cloak of k users can be composed. When all users are confirmed to be inside the toleration values of

their neighbors, the smallest area that includes all of the users is calculated. For example, if there are four users that are located at (0,0), (1,0), (0,1) and (1,1) coordinates and k is equal to four, then the observed events of them are kept as (pseudo name, [0, 1], [0, 1], [time period], C). Someone, who looks at one of the observed events, understands that a user was present inside the area, which has range of x values from 0 to 1 and y values from 0 to 1, at the specified time period.

Since the time gap between two events is defined as half an hour in our implementation, the toleration value in time space should be one less than quarter of an hour; because two consecutive events can collide. For example, an event occurs at 01.00 and the next one occurs at 01.30. If the toleration value in time space is 15, then time periods of two events will look like [00.45, 01.15] and [01.15, 01.45] after the application of K-anonymity. It is visible that the ending time of one observed event intersects with the beginning time of the next one, hence the toleration in time space could roughly be 14 minutes. We want to avoid intersection between consecutive events; because while evaluating our K-anonymity implementation, observed events are selected according to their order in time and if the wrong event is selected, then the location privacy values might be overestimated. For instance, the event that occurs at 01.00 is located at (0, -1) and observed at (-0.6, -0.8). The next event is located at (1, 0) and observed at (0.8, 0.6). Under normal circumstances, the distance between actual and observed events is 0.632 for both events that take place at 01.00 and 01.30. However if the observed events are confused as a result of time intersection, then the distance would be calculated as 1.788 for both cases, which leads to overestimation in the assessment of the protection mechanism.

It is mentioned that the evaluation method is adopted from Distortion-Based Metric [38], which is based on traces, and K-anonymity is not suitable for traces. Moreover, Distortion-Based Metric works with exact locations and K-anonymity works with areas. It was necessary to adapt our simplified K-anonymity implementation to Distortion-Based Metric so that we can evaluate its effectiveness.

First of all, we tried to build traces by looking at the observed events, when we simulated the adversary. If k value is equal to the number of users, then it appears as if there is only one trace in the system. In this case we look at the actual events of users and calculate the distance to the corresponding observed event, which is closest in time space. Then combining the distance with the probability distribution gives us the distortion that K-anonymity

provides. If the k value is less than the number of users, then there can be several traces. For example, if there are 4 users and 4 time instances and k is equal to 2, then there are at least 2 distinct observed events at each time instance, which leads to 16 ((2 distinct events)$^{4 \text{ time instances}}$) possible traces.

The incompatibility of location and area problem is solved by picking a random point inside the specified area. Selecting a random point in an area is definitely not the same as having an area and it might also lead to underestimation or overestimation in the location privacy results. However it is still a reasonable thing to do, as an adversary is able to look at the area from K-anonymity result and guess where the user might be in that area. It is possible to do smarter things here; but it probably requires more simulations and this is an option that we take according to the deadline of the project.

Chapter 9

Probability Assignment

Probability assignment is one of the most important factors in the location privacy problem. As it is explained in the introduction chapter, the adversary in the LBS is simulated in order to assess how much location privacy a user can experience. The adversary has knowledge of users that we cannot know to what extend. Adversary acquires observed events of users, generates possible traces out of these events and then he/she uses his/her knowledge to guess or predict which traces or events could belong to whom. Guessing traces or events is interpreted as assigning probabilities to them according to the knowledge of the adversary. For example, if the adversary is tracking a user and he/she knows that the user visits a location everyday at the same time, then he/she observes the events at the specified time and sees that there are four events but only one of them is at the expected location. In this case, adversary assigns 100% probability to that event and 0% to other three events. After having pinpointed an event that probably belongs to the user, adversary starts observing other events in order to form the probable trace of the user. Adversary, again, considers his/her knowledge of the user and the distances of the observed events, which occur at other time instances, to the pinpointed event. This process continues until all the observed events are considered. One could understand from this example that assessing a user's location privacy requires building the correct trace model, calculating distances between actual and observed events and applying probability distributions over the distance calculations. All of these factors are crucial, because if the trace model or the user's trace behavior is built incorrectly, or the probability distribution is chosen wrong, then the distance calculations are affected and if the distance calculations are wrong, then the resulting

location privacy values cannot reflect the real situation.

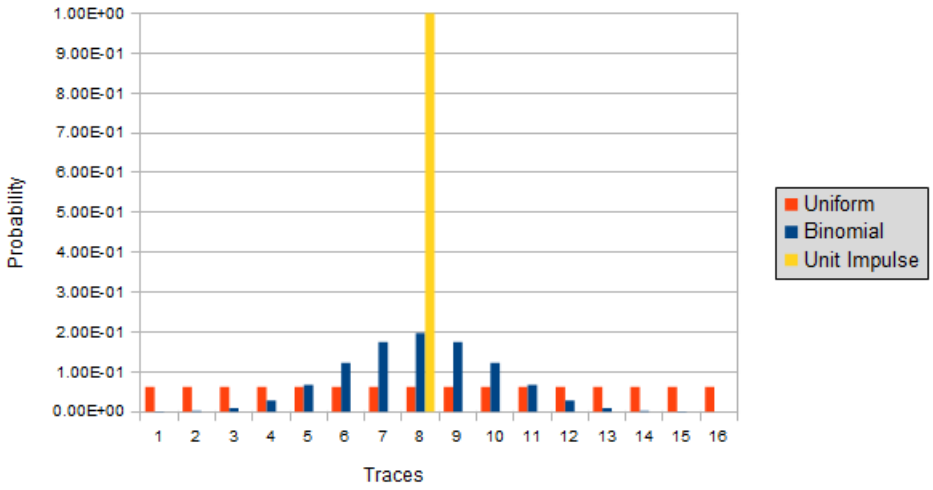


Figure 9.1: Probability distribution functions over 16 possible traces

We consider three different probability distribution functions, which are uniform, unit impulse and binomial (p=0.5), in this project. An example of each distribution over 16 possible traces is presented in Figure 9.1.

- When **uniform distribution** is considered, the adversary cannot distinguish a user from the others, hence all possibilities look equally probable. In Figure 9.1, every trace has 1/16 probability to belong to a specific user. An adversary has different probability distributions for each user, because he/she has different level of knowledge of each user. An adversary, who considers uniform distribution for tracking a user, is considered a weak adversary, because if the adversary has no knowledge of the user and acquires observed events, he/she can already say that every trace is probable to belong to the targeted user. Therefore, weak adversary cannot add more information to what he/she already has, or, in other words, he/she cannot decrease the uncertainty in the observed events/traces.

- In the case of **unit impulse**, the adversary is strong; because he/she can select a trace as the targeted user's trace with 100% certainty. This

is probably because of the knowledge that the adversary has about the user. It does not mean that adversary tries his/her luck to find the user and he/she gets it right in the first try. The adversary is absolutely sure of the trace that it belongs to the user. In Figure 9.1, targeted user's trace is numbered as 8. Since adversary assigns 100% probability to a trace and the sum of probabilities of all traces adds up to 100%, the rest of the traces have 0% probability, thus there is no yellow bar for other traces in Figure 9.1.

- The other probability distribution that we consider is the **binomial distribution** (p=0.5) and its strength lies between the weak and strong adversaries. For example, the adversary that uses binomial distribution thinks that the trace of the targeted user is numbered as 8; but he/she also thinks that traces that are numbered as 7 and 9 are also very likely. Therefore, the adversary does not have one choice as in the case of strong adversary; but he/she also does not think that all the possibilities are equally probable. He/she has a favorite trace, that is the one numbered as 8 and is the most probable trace that belongs to the user. He/she also thinks that there is no need to consider traces that are numbered as 1, 2, 14, 15 and 16.

In Figure 9.1, the ordering of traces is a significant detail. Sixteen traces, which are numbered, are placed on the x axis of the graph. The question is which number corresponds to which trace. If it is assumed that there are 2 users and 4 time instances in the scenario, then we have 16 (2^4) possible traces. For example, some of the traces could look like <a1, a2, a3, a4>, <b1, b2, b3, b4>, <a1, b2, a3, a4>, etc. The users are named as a and b, and the time instances are concatenated to the user names, e.g. a1 means user a at time instance 1. Since time instances in a trace are sorted in ascending order and cannot be repeated, each trace has to have events with 4 different time instances. One can guess that the trace that is numbered as 8 is either <a1, a2, a3, a4> or <b1, b2, b3, b4>, by looking at adversaries probability distributions in Figure 9.1. However the other traces are harder to guess by just observing the probability distributions, because, for instance, the adversary, whose knowledge corresponds to binomial distribution, believes that the traces that are numbered as 7 and 9 are also very probable traces for the targeted user. Even if we assume that the targeted user is, for example, a, we still have to consider distances between observed events. Traces that

are numbered as 7 and 9 have probably one different event in comparison to the actual trace; but in that case there are 4 possibilities such as <b1, a2, a3, a4>, <a1, b2, a3, a4>, <a1, a2, b3, a4> and <a1, a2, a3, b4>. As the number of traces increase and we consider the traces that are less likely for the targeted user, it gets harder to predict how the traces are ordered on the x axis. There might also be strange cases like a trace which has more events from user b and seems very likely for user a; because of short distances. For example, user b follows a trace that is very close to user a's trace and it takes a shortcut towards the same end point. In this case, the probability distribution(s) will be affected and the location privacy values might not reflect the reality, such that it provides an underestimation or an overestimation in the results. Thus the probability distributions are dependent on the ordering and the characteristics of traces, which means that they are also dependent on the distance metric.

As it is mentioned in the adversary part of the implementation chapter, two approaches for assigning probabilities to traces/events are tried. In the "exponential generation of traces", all of the traces are generated exponentially by searching among the observed events. For example, if there are 2 users and 4 time instances, the observed events could be seen in Table 9.1.

Time Instances	t_1	t_2	t_3	t_4
User a	a1	a2	a3	a4
User b	b1	b2	b3	b4

Table 9.1: 8 events of 2 users at 4 time instances

The process of exponential generation of traces start from first time instance, take an event and move on to next time instance. It is repeated until all the time instances and users are exhausted. For instance, generated traces are <a1, a2, a3, a4>, <a1, a2, a3, b4>, <a1, a2, b3, a4>, <a1, a2, b3, b4> and so on. When all of the traces are generated, the probabilities are assigned to them in HashMap<Trace,Float>. For the strong adversary, an example is <<a1,a2,a3,a4>,1.0>. When this approach is used, the mapping from traces to probabilities is done for only one user; hence, when the program is run to assess the location privacy that user a has, HashMap must be re-instantiated for the next user. At least, the generated traces are kept in a separate data structure, so that they can be reused for each user's location privacy assessment.

In the "no generation of traces" approach, we do not generate traces out of the observed events, instead, we look at the user and the time instance, and assign probabilities to different paths. The mapping is done from String to Float[] (array of Float), again by using a HashMap<>. An example for String object is "(0,0)", which means that "first user at first time instance". An example of mapping is <"(0,1)", [1.0, 0.0]>, which means that "first user chooses to follow the first path at the second time instance". This example is again from the strong adversary, as he/she knows the path that user a has followed.

The comparison of two approaches tells us that the outcome of mapping could be the same at different costs. In the first example, traces are generated once and probability assignments are renewed for each user. This approach is pretty simple to write and understand, however the time it takes to exponentially generate all traces grows extremely fast and forms a bottleneck in the performance of the application. It could be used for simple scenarios, in which there are few events; but definitely not for thousands or more events. On the other hand, there is no need to generate traces in the second approach, but the probability tables for each user become quite big as the number of events increase and it gets more complex, in the code, to handle all of those tables at one run. In other words, the second approach does not require renewal of probability assignments and rerun of the program for each user. Every user could be taken into consideration in an efficient manner at one run; however the program becomes too complicated and the time it takes to write and debug the code is longer than the first approach.

9.1 Adversary Modeling

The investigation on existing protocols showed that there are many variables in a system that tries to preserve the location privacy of users. The variation leads to many assumptions in every study. For example, the adversary model of the study is a variable. Some of the papers might not give away the details of their adversary model and others might consider simple models as their work is mostly on the protection mechanism. Building a realistic adversary is hard and takes too much time and effort.

In this project, we choose to consider 3 different adversary models, which are strong, weak and in between them. Strong adversary knows a lot about the users and has high certainty on users' traces. On the other hand weak

adversary knows nothing in addition to the observed events and has high uncertainty on traces of users. The adversary's knowledge of users is transformed into a probability distribution function that we explain previously. The probability distribution functions that we have considered are in some regular shape such that it fits a curve or a line. However it might not be that simple, because even if all traces look somehow probabilistically related to a user, an adversary might eliminate some of them according to his/her knowledge. In that case, a probability function might not reflect the adversary in the same way, as it is mentioned before. When the order and the number of traces change, the probability function also needs to be updated. It might be useful to combine several functions so that the probability function is resistant to changes or noises.

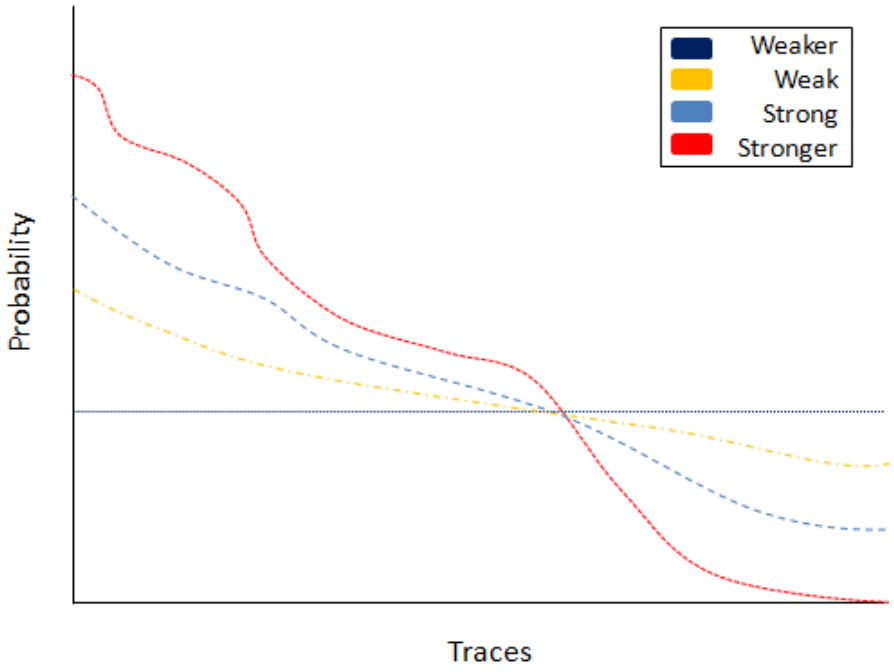


Figure 9.2: Adversary Modeling

In Figure 9.2, different adversary models are presented theoretically. In x axis, the traces are sorted according to their similarity to the actual trace in descending order. The weaker adversary model in Figure 9.2 corresponds to the uniform distribution. As the peak of the curves rises, strength of the

adversary increases or, in other words, knowledge of the adversary increases. Figure 9.2 is plotted for only one user. As adversary's knowledge of users change, the graphs also look different.

Chapter 10

Distance Metric

The distance metric is another crucial part of the location privacy evaluation of a user, as it has direct impact on the results. We have witnessed this impact at a stage of the project, because while trying to reflect Distortion-Based Metric [38] in the implementation, there was a misunderstanding of the distance measure that is mentioned in the paper. The purpose of this section is to share our experience of trying to understand the distance metric, which consumed too much time, and make the distance metric easier to understand.

The paper uses distance to tail for each trace; but the tail is not the tail of the actual trace. They consider each user and time instance, in order to figure out small traces, which is actually named Tpath [38] by them. Each small trace is part of the actual trace. They predict a location for user u before time t, make that location as the head of the trace and then consider a tail at time t. There is only one time difference between events, so the tail is the next event after the last prediction. However, this was not understood at the early stage of the project.

At first, there was nine events of three users at three time instances, which means 27 (3^3) traces, in our scenario. Having few events and traces did not make the problem very obvious. It could be motivated as the adversary knows the tail of the trace and tries to predict the trace according to the tail. However it was necessary to run simulations in order to understand the problem and the reason behind it. Nine events were input to the simulation and the adversary was defined as weak. In the scenario, there were three users, which are a, b and c. User a and b followed almost similar traces and their paths overlapped between two time instances. User c followed a completely different trace, which only crossed traces of user a and b perpen-

dicularly while they were moving together. Since the adversary is unable to distinguish between users and, users a and b followed similar traces, the expectation was user c would have the highest location privacy. The result was completely the opposite of the expectation. The reason user c had the lowest location privacy was locations of tail events of users a and b; because they were very close to the second event of c. Thus at every time instance user c experienced low distance to tail events.

Considering only the tail event was not enough, so in the next run, both head and tail events were introduced to the distance metric, as if the adversary knew them and tried to fill in between those events. While assessing the location privacy of users, at each time instance the distances from observed event to head and tail events were measured. The result made more sense than the previous one. The gap between location privacy values of three users was reduced; but it still was not as expected.

In the next run, three locations are considered at each time instance. These locations belonged to actual events at the beginning of the trace (head), at the time instance that is considered and at the end of the trace (tail). Result of this run met with our expectation; however the values were low, because of the aggregation of three different locations.

Finally, head and tail events of the actual trace are removed from the measurement, which means that the only distance at each time instance is the one that is between actual and observed events. By this way, the results were as expected and higher than the previous ones.

After completing all of these simulations, distance calculation in [38] is reconsidered and made complete sense. Thus, as it is mentioned in [38], all of the possible traces of all users are not considered at once while trying to evaluate the location privacy of a user. An event is selected as a starting point and the distance to other events at the same time instance are calculated. The distances are weighted in a probabilistic manner according to the adversary's knowledge of the specific user. This procedure is repeated for every time instance until a specific time t. The distance to tail, which is included in [38], means the distance to the event at the next time instance, as the time instances are incremented by one at each iteration of the algorithm [38]. Therefore, the observed trace of a user is evaluated as a sequence of atomic distance calculations at each time instance.

Chapter 11

Results

11.1 K-Anonymity Results

In the simulations, the number of users and time instances, sequentially, are defined as 12 and 10. As the number of users are defined as 12, k values that are considered are 2, 3, 4, 6 and 12, in order to serve all of the users with K-anonymity. If k value is not a multiple of users, then some of the users are left out alone, as K-anonymity works on a first come first serve basis. For example, if we use k equals to five, then ten users are served in groups of two and the remaining two users have to try their luck with k equals one, which means no cloaking. They cannot be served in a group of five, because the events that come after those two belong to next time instance and the toleration in time space does not allow cloaking with them.

The simulations are run for 30 times, because of 5 different k values, 3 different trace models and 2 different adversary models. For each run of the simulation, the program is automatically repeated 10 times; because the traces are generated randomly. Even if the trace model is defined for each run, the locations of events are randomly assigned. Since we do not want to suffer from an extreme spread of locations, we run the tests 10 times all over again. Very closely or widely spread events are two examples of extreme cases. If all of the events are located very closely, then the cloaking area would be very small leading to very low location privacy values. On the other hand, if all of the events are spread in a very wide area, then the cloaking area would be very big leading to very high location privacy values.

3600 (12 users * run for 10 times * 5 different k values * 3 different

trace models * 2 different adversary models) results are obtained from the simulations of our simplified K-anonymity implementation. The simulation results are consistent with the papers on K-anonymity; such that when k value increases, the location privacy values, which users can expect, also increases. This is because of the widened area in order to cloak all 12 users at once. The automated generation of traces gives well spread locations most of the time as we use uniform distribution for the randomness of them. As the points are uniformly distributed, one can imagine that the area, which covers two points, is smaller than the area, which covers twelve points. Even if we pick a random point within an area, we can see the increasing trend for increasing k value.

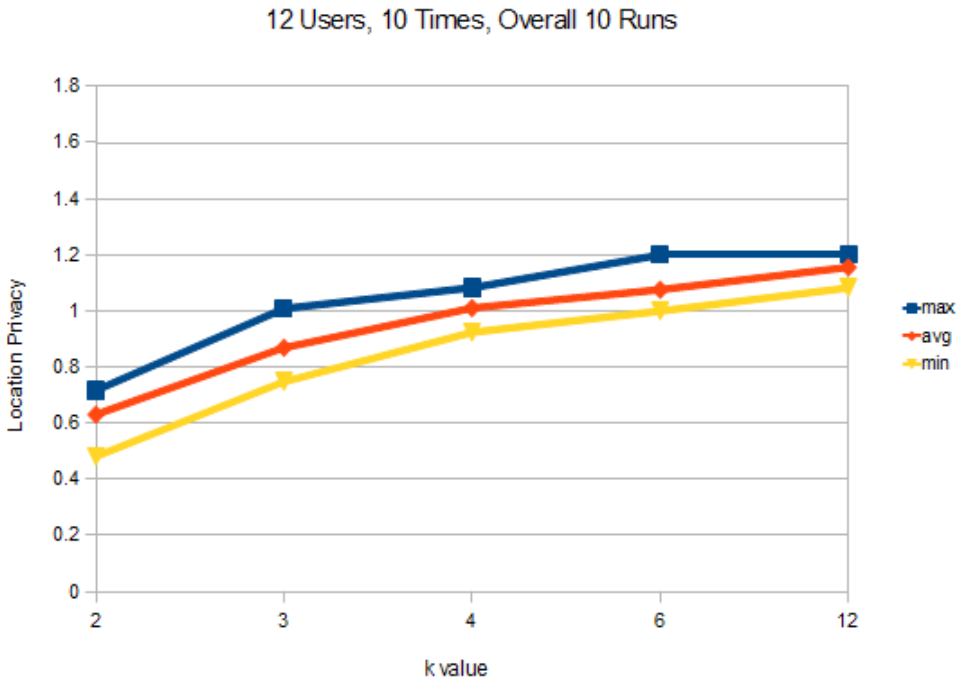


Figure 11.1: Statistical information, such as maximum (blue line), average (orange line) and minimum (yellow line), from simulations of circular traces and strong adversary

In Figure 11.1, statistical information, such as maximum, average and minimum location privacy values, is extracted from the simulation results of simple K-anonymity implementation. In this simulation, circular traces of

users are generated for 12 users and 10 time instances, which means that there are 120 events in the simulation. The points in the graph could be interpreted as the maximum, average and minimum location privacy values when k equals to an integer value. For example, when k equals to 2, maximum, average and minimum location privacy values are, sequentially, 0.717, 0.631 and 0.481. When k equals to 12, the values are 1.2, 1.16 and 1.08. These values are extracted from average values of 12 users overall 10 runs. Thus one can imagine that there are even higher values than the maximum and lower values than the minimum among the simulation results, but those values are caused by random generation of actual events. Therefore, average location privacy values of each user is extracted at first and then the result is statistically analyzed.

The adversary model that is used in the simulation of Figure 11.1 is strong adversary, who knows or guesses exactly in which cloak a user is hidden. That is to say, even if there are several distinct cloaks at a time instance, the adversary only considers the cloak, in which the targeted user is protected. For example, when k equals to six, there are two cloaks at each time instance and if a user is placed in the first cloak, the adversary considers that cloak in order to reach the user. In technical terms, as distortion is calculated by multiplying the distance and the assigned probability, the expected distortion, which the adversary experiences for tracking the user, is only the distance between actual and observed events of the user; because the probability is 1 for all observed events of the user. Since expected distortion is calculated for a specific time, the location privacy of the user is calculated by taking the average of all expected distortions at all time instances. [38]

The location privacy values are plotted in y axis of Figure 11.1. The location privacy values that are plotted on the graph are dependent on the trace model and not normalized. The values are dependent on the trace model, because the events are plotted on the unit circle, which is a restricted area that covers a maximum distance of 2. If this model is applied to a real world scenario, the circle needs to be expanded to fit the area. In that case, the results are again dependent on the scenario, as the distance between events are considered to calculate the distortion and, of course, the location privacy of the user. If we were to normalize the location privacy values, it would have been dependent on the distance unit. In our case, the normalization factor could be slightly greater than 2, because the longest distance on the circle is 2, however the smallest box that contains the actual events might go beyond the boundaries of the circle. It is still not a very important detail

in this case, because the increasing trend of the location privacy (on y axis) according to the increasing k value (on x axis) is present regardless of the normalization. Increasing trend is visible because as k value increases, more users are cloaked together and the area of the cloaks gets larger in order to include more users inside them. When the area gets larger, randomly picked observed events lead to increased distance to actual events of users on average. There might be cases in which distances to observed events are short or long, however the important thing is to see how it scales in many runs.

The simulation, which provides the results that are presented in Figure 11.1, is also run for cross and parallel trace models. The results show that location privacy is improved when more users are cloaked together regardless of the trace model, however the amount of increase and the location privacy values are different from Figure 11.1.

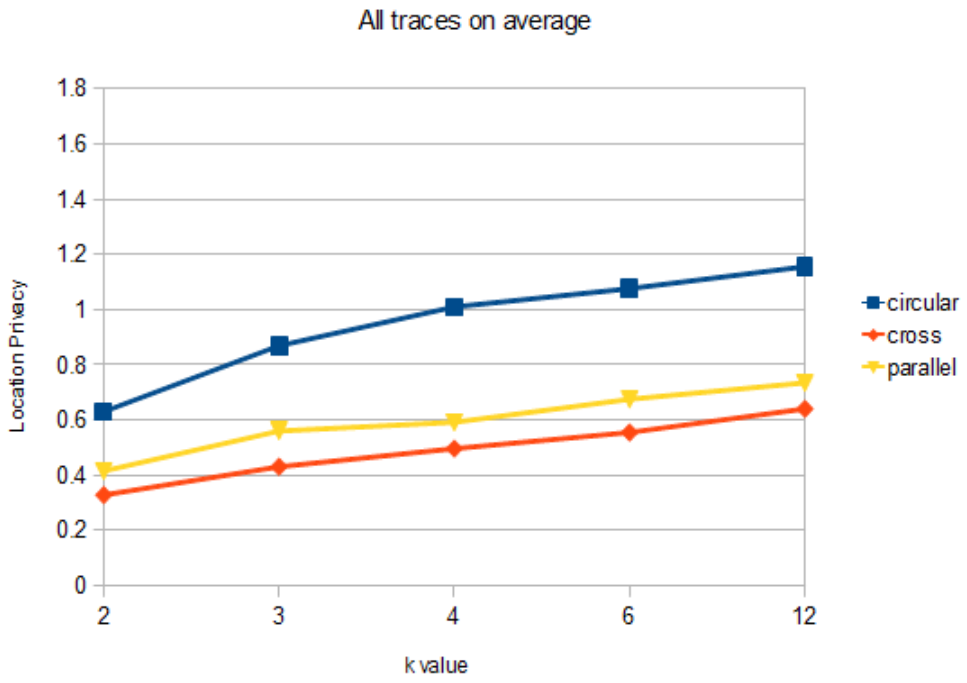


Figure 11.2: Average location privacy achieved in different trace models when adversary is strong. Circular traces (blue line), cross traces (orange line) and parallel traces (yellow line)

In Figure 11.2, average location privacy values that are achieved by con-

sidering different trace models are presented. When a user crosses another user's trace, the location privacy values that each user experiences, on average, is the lowest in comparison to parallel and circular traces; because users get closer to each other while crossing each other and when they do, they cause cloaking boxes to be very small in area. When cloaking boxes get smaller, distances between actual and observed events become shorter which leads to small distortion and in return less location privacy. When a user follows a trace, which is parallel to other users' traces, he/she stays distant to other users. In this case, the cloaking box can cover larger area, in which users could enjoy higher location privacy values. However, the users are still located inside the circle most of the time, thus the location privacy values are a little bit higher than the ones in cross traces. The highest location privacy values are achieved in circular traces that are composed of events which are located on the circle. When actual events are located at the edge of the circle, the distances between actual and observed events could be maximized leading to high distortion in the eye of the adversary. The reason for observing greater amount of increase in plotting of circular traces might be the independence of starting at any point on the circle. The points are randomly distributed among 360 degrees for circular traces whereas they are distributed among 180 degrees for cross and parallel traces. Taking paths inside the circle is much restricted than wandering around the circle.

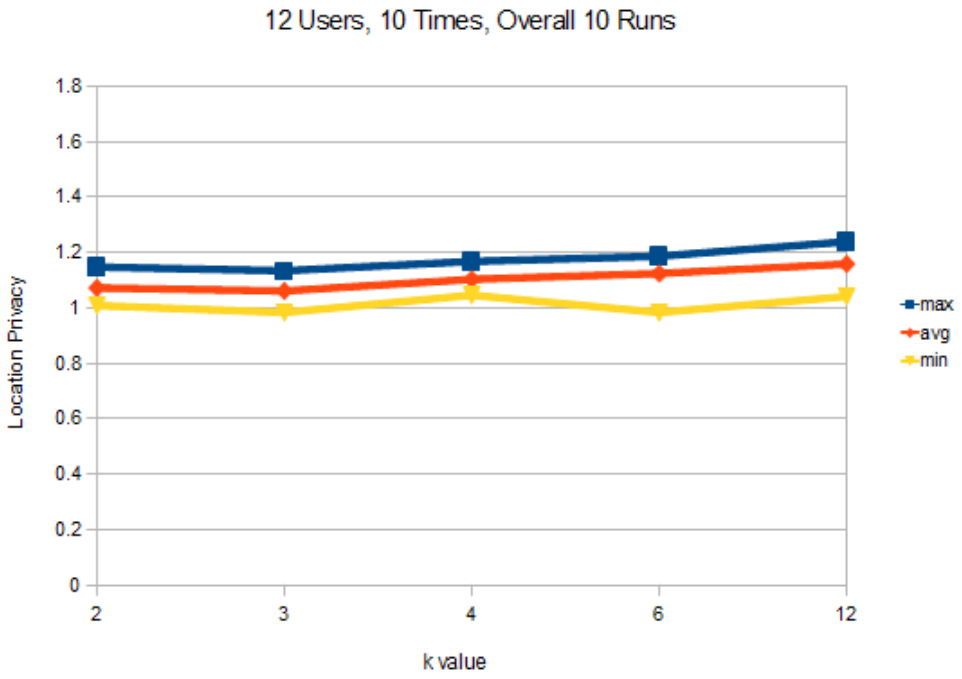


Figure 11.3: Statistical information, such as maximum (blue line), average (orange line) and minimum (yellow line), from simulations of circular traces and weak adversary

The only difference between Figure 11.1 and 11.3 is the adversary model. In Figure 11.1, the adversary is able to track the user successfully. The only thing that prevents the adversary from exactly locating the targeted user is the distortion that is caused by the cloaking box of K-anonymity. In Figure 11.3, the adversary is weak meaning that he/she cannot distinguish a user from the other ones. In this case, each user has (1/number of users) chance of being the targeted one by the adversary. Therefore, the adversary needs to consider every cloak at each time instance. Distortion, or in other words adversary's expected error in finding a specific user, is calculated by multiplying the distance of cloaks to the actual event of the user with the probability of the user, which is 1/12 in our scenario. When Figure 11.1 and 11.3 are compared, it is obvious that users enjoy higher location privacy as the strength of the adversary weakens. One of the important details in Figure 11.3 is the location privacy values that are achieved when k equals

to 12. Since there is only one cloak when k value equals to the number of users, the difference between the strong and weak adversaries disappears. Furthermore, Figure 11.3 reflects the claim of Shokri et al. [38] about K-anonymity, which means that the adversary's probability assignment of users is not visible when K-anonymity is applied; because all of the observed events of the users look equally probable. They also say that K- anonymity provide somewhat constant level of location privacy independent of the adversary's choices. One can see that when all of the observed events are treated equally, the level of location privacy does not change much. Shokri et al. thinks that the adversary might be able to distinguish a user by looking at different parameters of the observed events and/or having extra knowledge about the user, but this is something that Figure 11.3 cannot reflect.

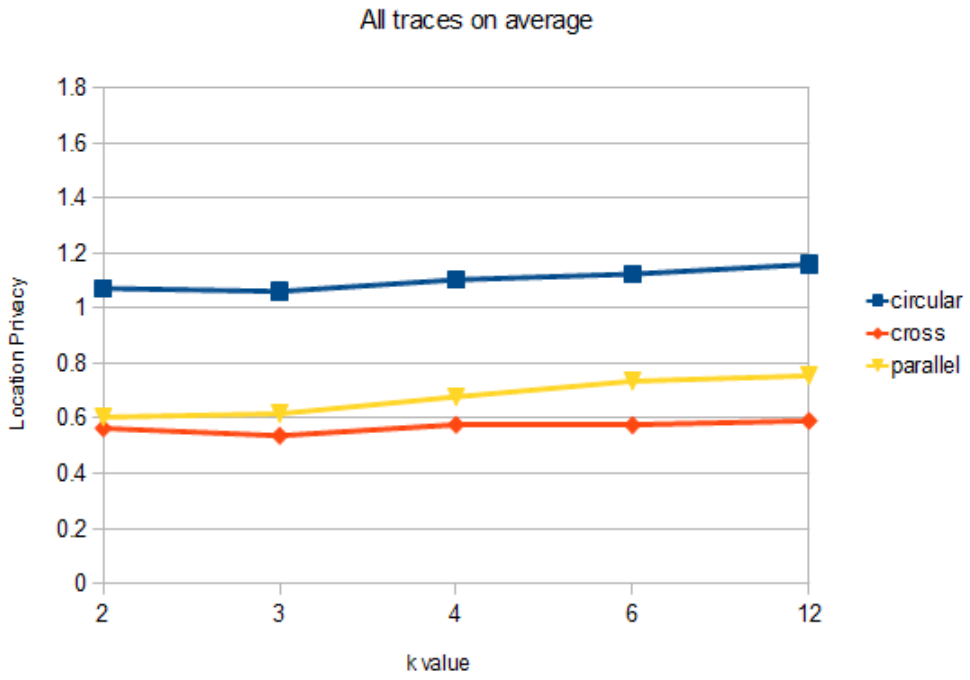


Figure 11.4: Average location privacy achieved in different trace models when adversary is weak. Circular traces (blue line), cross traces (orange line) and parallel traces (yellow line)

It is, again, possible to see different levels of location privacy achieved in different trace models when the adversary seems weak. A careful reader

might notice that the orange line in Figure 11.3 is the blue line in Figure 11.4. In the same way, one could imagine graphs of cross and parallel traces very similar to Figure 11.3; but having lower location privacy values and slight increase in them as k value increases.

11.2 The Performance Analysis of Automated Generation of Traces

The automated generation of traces are tested for various numbers of users and time instances. The results of the tests will be presented and explained in this section.

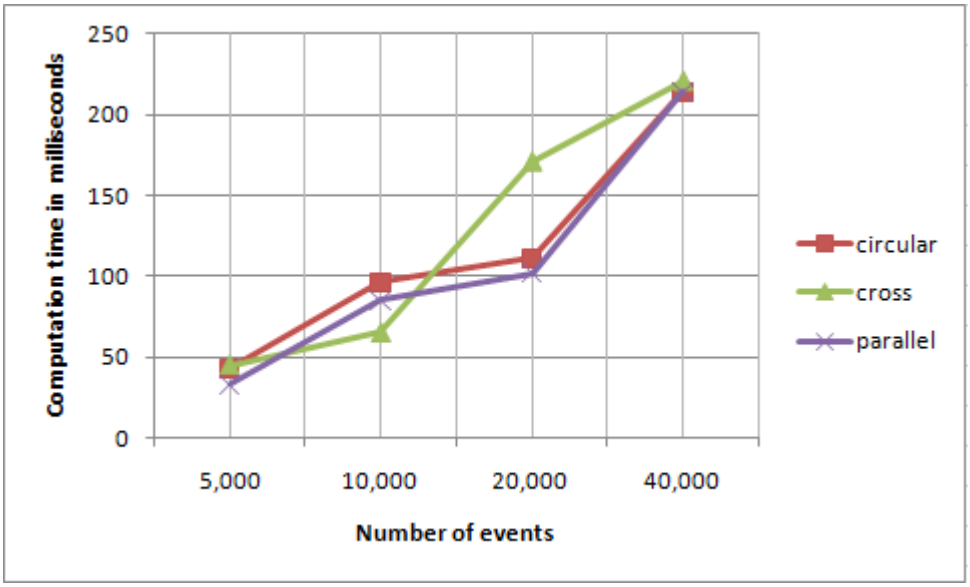


Figure 11.5: Performance graph of generated events of 100 users for 50, 100, 200 and 400 time instances.

In Figure 11.5, there are 100 users and events of each user is generated for 50, 100, 200 and 400 time instances. Number of events equals to number of users multiplied by number of time instances. Number of traces equals to number of users as each user has one actual trace, which is a sequence of events for the number of time instances specified. The computation time that it takes to generate events is visible on y axis of the graph. Generating 40 000 events takes even less than one quarter of a second, which might be regarded as a reasonable performance. The reason why cross traces has a different curve in Figure 11.5 might be because of different calculations for each trace model. For example, cross traces require much more "if" checks and computations, which require method calls from Math library of Java [8], in comparison to other trace models.

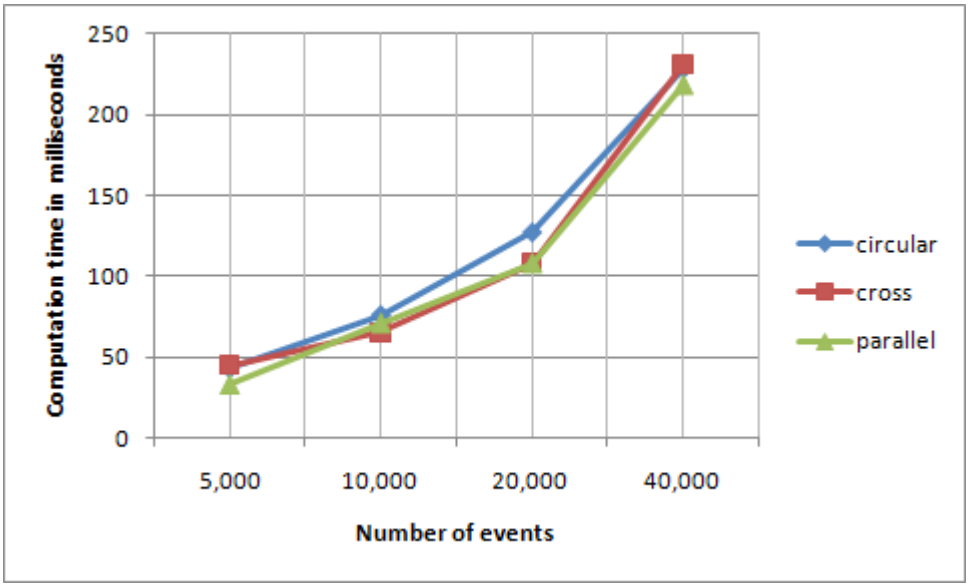


Figure 11.6: Performance graph of generated events of 100, 200, 400 and 800 users for 50 time instances

In Figure 11.6, the number of events are kept the same as in Figure 11.5; however, in this graph, the number of time instances are kept the same (50) and the number of users are doubled at each step (100, 200, 400 and 800). Even if the values are almost the same in Figures 11.5 and 11.6, there is almost linearly increasing trend in Figure 11.6, where as in Figure 11.5 there is not. Probable reason for observing this difference between two graphs is basic dependency of generation algorithms on the number of users. All of the algorithms start with a for loop, that iterates for each user, so that starting point of each user is selected. The number of times is considered inside that for loop. Thus, the computation time of generation of traces is linearly dependent on the number of users.

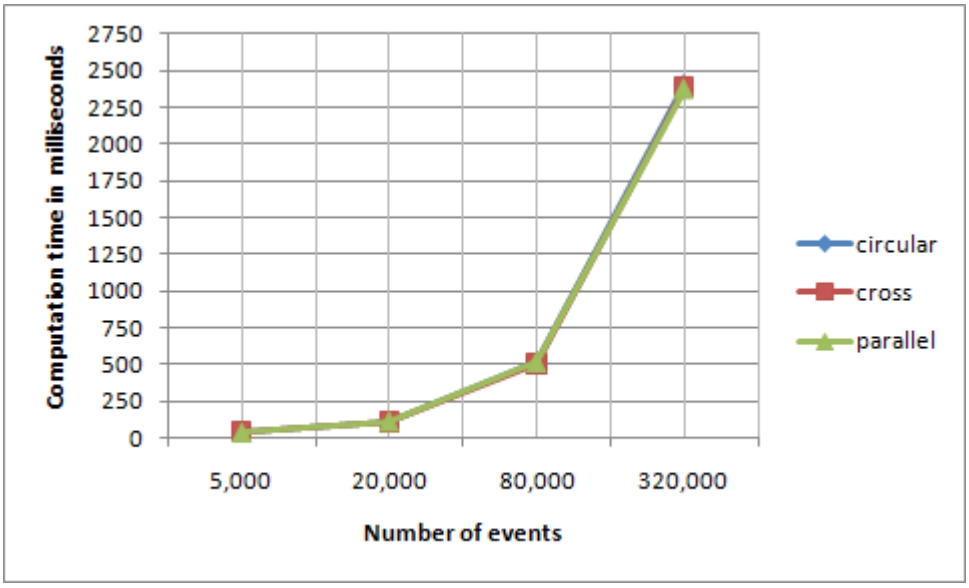


Figure 11.7: Performance graph of generated events of doubled users and time instances (100 users, 50 times), (200 users, 100 times), (400 users, 200 times) and (800 users, 400 times)

In Figure 11.7, the number of users starts from 100 and the number of time instances start from 50. Both the number of users and time instances are doubled at each run. The faster the number of events increase, the faster the computation time rises. The maximum number of events that are generated using the abstract model is 320,000 and the process takes about 2400 milliseconds.

Chapter 12

Conclusion

In this project, we have investigated 22 papers on location privacy. As a result of our study of these papers, we notice that there are approaches from different layers of communication stack and architectures. Moreover, some of the studies try to protect traces of users, whereas some others try to protect users at a location and time by forming cloaks of them or altering users' location and time information.

According to our analysis of related work on the location privacy, we decided to implement the location privacy evaluation model of Distortion-Based Metric [38], which we used to assess our implementation of K-anonymity solution. The modifications that we have done on K-anonymity implementation of [20] were elimination of personalization and adaptation to the evaluation model of Distortion-Based Metric. We have eliminated personalization from K-anonymity; because we aimed to observe results of K-anonymity protocol when it covers k-many users at a time, hence we made it work in all cases. If K-anonymity works in all cases, then personalization factor might not be a very interesting feature. By this way, we wanted to see best results one can achieve with K-anonymity. We assessed the location privacy values of the simplified K-anonymity implementation after adaptation to the Distortion-Based Metric; because we wanted to check validity of the claims made in [39]. The results confirmed both the claims made in [39] and the general view, which is location privacy increases when k value increases [20], on K-anonymity.

Furthermore, we implemented three different abstract models, which are circular, cross and parallel trace models, for generating traces of users. The events/traces that are generated by the algorithm is tested, evaluated for

performance and then used in the simulations of the simple K-anonymity implementation.

We also considered and examined effects of essential aspects, such as different adversary models and distance metrics, in the simulations. These parts have direct impact, which must be well understood and motivated in a similar study, on the location privacy results of the simulations.

Chapter 13

Contributions of the Thesis

There could be three contributions of this master thesis in comparison to other works on similar problems and these contributions could be mentioned as generation of events/traces of users based on abstract model, adaptation of a simple K-anonymity implementation to Distortion-Based Metric and consideration of different probability distribution functions for modeling adversary. After looking at many papers on location privacy, we decided that we could consider these three aspects in order to gain more insight about the problem of location privacy.

Generation of events/traces of users based on abstract model could be useful because the simulations show that specific traces of users match specific patterns when the location privacy values are plotted on a graph. There might be differences between the values that are computed from realistic scenarios and the scenarios that are created according to the abstract model; but the difference might not be big and the pattern might still be notable at the end. Most of the works on location privacy consider few users and events so that their evaluation could be done efficiently within the scope of a ten page paper. Since the abstract model automates the generation of events of users and it works quickly, one can generate traces in any shape easily. One can also look at different scenarios, in which there are different numbers of users and events. After having generated events of users, the emphasis could be on protecting those events and evaluating the protection mechanism.

As we have discussed in related work section, K-anonymity is not suitable for protecting traces of users, hence, it is hard to measure the location privacy that K-anonymity offers by using a metric, which considers traces of users. The ways to measure effectiveness of K-anonymity are looking at how

many queries are cloaked, how many anonymity levels, spatial and temporal resolution are achieved. [21] Since we want to evaluate different location privacy mechanisms in a similar way, we modified the output of a simplified K-anonymity implementation so that it can be evaluated as in the Distortion-Based Metric. Adaptation of a simplified K-anonymity implementation to Distortion-Based Metric helps to understand the problem of providing location privacy to users in a better way, because even if two solutions take different measures and provide different results they aim to address the same problem. It might be useful to be able to compare these two solutions in order to, at least, gain more insight about the location privacy problem.

Consideration of different probability distribution functions for modeling adversary is another interesting aspect of the location privacy problem. Since defining an adversary is a very hard task, considering different probability distribution functions for adversary let us learn more about how to provide location privacy to users. When a specific probability distribution function is selected, one might observe how the location privacy of a user changes, how effective the protection mechanism is, what kind of user patterns the distribution matches and if it could be possible to take precaution in the protection mechanism for such distribution. These questions could be answered by analyzing actual and observed events of users and the location privacy distribution among users. For example, we observed that when binomial distribution is used, the location privacy values of users are spread more closely that leads to smoother curves when plotting on a graph. On the other hand, when unit impulse function is used, the plotting of the location privacy values of users is composed of sharp rises and drops. Moreover, when unit impulse function is applied, the variation in the results is higher in comparison to the application of binomial distribution.

One more contribution of this project might also be helping students, who aim to work on location privacy, by showing them a clear analysis of the problem. There are several strong papers, which are well structured and motivated, on the location privacy subject. During this project, we studied 22 papers on location privacy and there are at least 10 more interesting papers that are noticed. We took note of interesting ideas from different papers and also tried to be creative while working on the code. We have worked on unclear or open parts from the papers that we have read and also considered most of the varying and, at the same time, essential parts of the location privacy problem. The work done in this project could be very useful for people who are introduced to the location privacy problem newly, because

most of their questions on the subject would probably be answered by our study and also they would have a wide range of possibilities for selecting a part of the problem that they could study.

Chapter 14

Future Work

This master thesis is only a starting point to study the location privacy problem. There are still many aspects to consider. Some of the future work that we consider are building more advanced adversary and trace models, developing an application in which several LPPM could be combined, considering distributed architectures and upgrading the single threaded implementation to a multi-threaded one.

The adversary models that we have considered so far are not very complex. Building an adversary model for the location privacy applications would be a very interesting problem to look at. It could be a very hard problem, however the adversary could be built using combination of different probability distributions.

More advanced trace models could be built and assessed. We have only covered basic traces of user that are cross, parallel and circular traces. One could look at different shapes of traces such as ellipse or parallelogram or an irregular shape. Traces of the users probably depend on the environments that users are visiting. Different cities have different arrangement of roads and structures that might affect the traces of users. These details could be considered while building trace models for users.

Combination of different location privacy metrics into one application using fuzzy logic could be another interesting study. We aimed to achieve this; but unfortunately limited time and complexity of the subject did not allow us. Each existing protocol on location privacy tries to address a different aspect of the same problem, which is providing location privacy to users querying the LBS, and has different complexity, strengths and weaknesses in comparison to the other ones. In some cases, it might be meaningful to use protocols

that consider traceability, whereas in another case K-anonymity might be an efficient and satisfactory solution. The fuzzy logic is necessary for understanding the scenario, so that proper protection mechanism(s) could be taken. The situation a user is in could be assessed according to the gathered information from the users and recommendation could be made to the user or it could be directly applied. A unified evaluation framework for location privacy is probably needed for such an application. Different location privacy mechanisms could be compared in effectiveness under certain circumstances. This is similar to our adaptation of K-anonymity to Distortion-Based Metric in order to assess the location privacy of users. Other location privacy solutions could be adapted as well, so that they could be compared not just on paper but in an application, hence they could be combined in one application to work together simultaneously.

The focus of this project stayed on centralized architecture, because of time constraint and wide range of details in the problem. There are also distributed architectures for location based services. One could investigate them in order to understand their strengths and weaknesses, and also compare with centralized architecture. There could be things that are important in a centralized architecture but not in distributed ones or vice versa.

The implementation so far is run using a single thread in this project, as the CPU of the machine that is used for it is single core. A machine with multiple core CPU could be used and different parts of the application could be implemented in a multi-threaded way. One benefit of this modification could be speeding up the application so that more complex scenarios, mechanisms and evaluation methods could be used. For example, now, every component of the system run in sequence in the order of generate events, apply location privacy preserving mechanism (LPPM), simulate adversary and assess location privacy. Generation of actual events can be done on one thread and as this thread works actual events can be handed to the LPPM, which can run on another thread and place observable events in a queue or buffer that the adversary is watching on another thread. The assessment mechanism can run on a different thread and assess adversarys success rate in time. There might be another benefit of the modification, which leads to interesting results such as adversary might not be able to deduce much until a point in time; but after that point he/she might learn about a user instantly. The implementation that is done for this project can be seen as an offline analysis of the situation, because all of the actual events are transformed into observable events and then adversary observes them and then

only we can assess the situation. If a multi-threaded implementation could be developed, it would be possible to monitor the adversary's view of users in time. It might not be possible for the adversary to learn about a user in a short time.

Bibliography

[1] Aroundme, http://www.tweakersoft.com/aroundme.html, 2011.

[2] Eclipse, http://www.eclipse.org, 2011.

[3] Global positioning system, http://en.wikipedia.org/wiki/gps, 2011.

[4] Gnuplot, http://www.gnuplot.info, 2011.

[5] Google latitude, https://www.google.com/latitude/, 2011.

[6] Google maps, http://maps.google.com, 2011.

[7] Google scholar, http://scholar.google.com, 2011.

[8] Java, http://java.sun.com, 2011.

[9] Openoffice.org, http://www.openoffice.org, 2011.

[10] Wireless lan, http://en.wikipedia.org/wiki/wlan, 2011.

[11] C. A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. Location privacy protection through obfuscation-based techniques. In *Proceedings of the 21st annual IFIP WG 11.3 working conference on Data and applications security*, pages 47–60, Berlin, Heidelberg, 2007. Springer-Verlag.

[12] Alastair R. Beresford and Frank Stajano. Mix zones: User privacy in location-aware services. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, PERCOMW '04, pages 127–, Washington, DC, USA, 2004. IEEE Computer Society.

[13] Ling Liu Bugra Gedik. A customizable k-anonymity model for protecting location privacy. Technical Report GIT-CERCS-04-15, Georgia Institute of Technology, April 2004.

[14] Chi-Yin Chow, Mohamed F. Mokbel, and Xuan 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, pages 171–178, New York, NY, USA, 2006. ACM.

[15] Richard Chow and Philippe Golle. Faking contextual data for fun, profit, and privacy. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, WPES '09, pages 105–108, New York, NY, USA, 2009. ACM.

[16] Dan Cvrcek, Marek Kumpost, Vashek Matyas, and George Danezis. A study on the value of location privacy. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, WPES '06, pages 109–118, New York, NY, USA, 2006. ACM.

[17] Matt Duckham and Lars Kulik. A formal model of obfuscation and negotiation for location privacy. In *Pervasive Computing*, pages 152–170, 2005.

[18] Hannes Federrath, Anja Jerichow, and Andreas Pfitzmann. Mixes in mobile communication systems: Location management with privacy. In *Proceedings of the First International Workshop on Information Hiding*, pages 121–135, London, UK, 1996. Springer-Verlag.

[19] Lars Fischer, Stefan Katzenbeisser, and Claudia Eckert. Measuring unlinkability revisited. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, WPES '08, pages 105–110, New York, NY, USA, 2008. ACM.

[20] Bugra Gedik and Ling Liu. Location privacy in mobile systems: A personalized anonymization model. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, ICDCS '05, pages 620–629, Washington, DC, USA, 2005. IEEE Computer Society.

[21] Buğra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7:1–18, January 2008.

[22] Marco Gruteser and Dirk 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, pages 31–42, New York, NY, USA, 2003. ACM.

[23] Marco Gruteser and Dirk Grunwald. Enhancing location privacy in wireless lan through disposable interface identifiers: a quantitative analysis. *Mob. Netw. Appl.*, 10:315–325, June 2005.

[24] Baik Hoh and Marco Gruteser. Protecting location privacy through path confusion. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 194–205, Washington, DC, USA, 2005. IEEE Computer Society.

[25] Baik Hoh, Marco Gruteser, Ryan Herring, Jeff Ban, Daniel Work, Juan-Carlos Herrera, Alexandre M. Bayen, Murali Annavaram, and Quinn Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *Proceeding of the 6th international conference on Mobile systems, applications, and services*, MobiSys '08, pages 15–28, New York, NY, USA, 2008. ACM.

[26] Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, pages 161–171, New York, NY, USA, 2007. ACM.

[27] Leping Huang, Hiroshi Yamane, Kanta Matsuura, and Kaoru Sezaki. Silent cascade: Enhancing location privacy without communication qos degradation. In John A. Clark, Richard F. Paige, Fiona Polack, and Phillip J. Brooke, editors, *Security in Pervasive Computing, Third International Conference, SPC 2006, York, UK, April 18-21, 2006, Proceedings*, volume 3934 of *Lecture Notes in Computer Science*, pages 165–180. Springer, 2006.

[28] Tao Jiang, Helen J. Wang, and Yih-Chun Hu. Preserving location privacy in wireless lans. In *Proceedings of the 5th international conference*

on Mobile systems, applications and services, MobiSys '07, pages 246–257, New York, NY, USA, 2007. ACM.

[29] Waseem Karim. The Privacy Implications of Personal Locators: Why You Should Think Twice Before Voluntarily Availing Yourself to GPS Monitoring. *Washington University Journal of Law and Policy*, 14:485–515, 2004.

[30] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. An anonymous communication technique using dummies for location-based services. In *International Conference on Pervasive Services*.

[31] Jiejun Kong and Xiaoyan Hong. Anodr: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '03, pages 291–302, New York, NY, USA, 2003. ACM.

[32] John Krumm. A survey of computational location privacy. *Personal Ubiquitous Comput.*, 13:391–399, August 2009.

[33] Hua Lu, Christian S. Jensen, and Man Lung Yiu. Pad: privacy-area aware, dummy-based location privacy in mobile services. In *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, MobiDE '08, pages 16–23, New York, NY, USA, 2008. ACM.

[34] Joseph Meyerowitz and Romit Roy Choudhury. Hiding stars with fireworks: location privacy through camouflage. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, MobiCom '09, pages 345–356, New York, NY, USA, 2009. ACM.

[35] Robert P. Minch. Privacy issues in location-aware mobile devices. In *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 5 - Volume 5*, pages 50127.2–, Washington, DC, USA, 2004. IEEE Computer Society.

[36] K. Sampigethaya, Mingyan Li, Leping Huang, and R. Poovendran. Amoeba: Robust location privacy scheme for vanet. *Selected Areas in Communications, IEEE Journal on*, 25(8):1569 –1589, oct. 2007.

[37] Reza Shokri, Julien Freudiger, and Jean-Pierre Hubaux. A Unified Framework for Location Privacy. Technical report, 2010.

[38] Reza Shokri, Julien Freudiger, Murtuza Jadliwala, and Jean-Pierre Hubaux. A distortion-based metric for location privacy. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, WPES '09, pages 21–30, New York, NY, USA, 2009. ACM.

[39] Reza Shokri, Carmela Troncoso, Claudia Diaz, Julien Freudiger, and Jean-Pierre Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*, WPES '10, pages 115–118, New York, NY, USA, 2010. ACM.

[40] Einar Snekkenes. Concepts for personal location privacy policies. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, EC '01, pages 48–57, New York, NY, USA, 2001. ACM.

[41] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10:571–588, October 2002.

[42] Latanya Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10:557–570, October 2002.

[43] Man Lung Yiu, Christian S. Jensen, Xuegang Huang, and Hua Lu. Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, pages 366–375, Washington, DC, USA, 2008. IEEE Computer Society.

[44] Tun-Hao You, Wen-Chih Peng, and Wang-Chien Lee. Protecting moving trajectories with dummies. In *Proceedings of the 2007 International Conference on Mobile Data Management*, pages 278–282, Washington, DC, USA, 2007. IEEE Computer Society.

[45] Ge Zhong and Urs Hengartner. Toward a distributed k-anonymity protocol for location privacy. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, WPES '08, pages 33–38, New York, NY, USA, 2008. ACM.