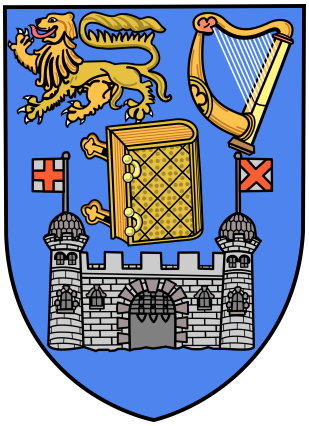
Privacy-aware Mechanism for Location-based Social Networks

A thesis submitted in fulfilment of the requirements for the M.Sc degree



Author: Guojun Qin

November 19, 2017

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Signed:

_____

Date:

_____

Summary

Recently, location-based social networks (LBSNs), such as Foursquare, Facebook, and Plenty of Fish, have attracted millions of users by helping them to build their social contacts and share useful information. LBSNs provide fine-grained and personalized services to their users, such as the 'location check-in' feature in check-in applications to obtain rewards, the 'people nearby' feature in dating applications for meeting interested people nearby, and the 'friends alert' feature in proximity notification applications for receiving a notice when a friend is close by. Users can use these features to do activities such as a) update their friends on their whereabouts; b) discover the best places to eat, drink, shop, or visit in a certain area; c) find new friends with similar interests or a date with a well matched person; and d) easily organize social activities. However, location sharing is a double-edged sword, which can on the one hand make life convenient but also might reveal private location information to curious application servers or malicious users.

In state-of-the-art location privacy mechanisms, some only provide location privacy protection against malicious users but not curious application servers. While some of the existing solutions obfuscate the actual location of their users, such as cell-grid based approaches, they suffer from false positives/negatives in proximity estimation. Some other solutions implement stronger security, i.e., location tags, to protect users' location privacy. However, these techniques usually rely on additional sniffing devices that are costly and cause an excessive drain on the battery of the users' handsets. The existing mechanisms are only designed to address privacy issues for specific types of applications in LBSNs and all suffer from different types of privacy attacks.

Our research contributions address four aspects relating to location privacy:

1. We address issues of privacy in LBSNs by protecting user's privacy against both application servers and malicious users; our approach employs encrypted cell-tower identifier sets instead of location coordinates for location proximity testing, which protects users' actual locations from being revealed to either malicious users or

application servers.

2. We create a privacy-aware mechanism that satisfies all categories of LBSN applications. For instance, 'k-anonymity' is designed to hide users' identities and actual locations when they send queries to fetch information from servers. This approach could be used in check-in applications but not dating and proximity applications that need to show users' identities. Our approach doesn't share actual location coordinates with any party but allows users or application servers to achieve application functions related to location.

3. We employ cell-tower identifiers as location tags, which can be directly accessed by mobile phones without a sniffing tool. This solution is able to resist more privacy attacks with an encrypted dataset and is more mobile friendly. Cell towers are distributed according to the density of mobile users, and their coverage ranges are also adjusted according to the density of mobile users. This means that a group of cell towers that covers a mobile user would dynamically shape a particular obfuscated region for each different location to which the mobile users move. Hence, we adopt this feature to provide a self-organizing location obfuscation solution to ensure users' location information is protected in LBSN applications.

4. We introduce the k-combination approach which is a more accurate proximity testing mechanis than k-shingling approach presented in (Zheng et al. 2012). Our approach takes all similar elements between two data sets into account when comparing the similarity of those data sets. The experiments in section 4.4.2 show that the results using our k-combination approach have better accuracy when compared with data using the k-shingling approach.

Acknowledgements

This work was supported by Science Foundation Ireland under the Principal Investigator research program 10/IN.1/I2980 “Self-organizing Architectures for Autonomic Management of Smart Cities”.

Contents

Chapter 1

Introduction

Mobile social network services have been used to improve citizens' lives in many areas, such as healthcare, public safety, social services, traffic, education, retail, economic development, rail, energy, utilities, airports and communications. However, since some sensitive personal data such as location and images might be collected as required during the service, maintaining privacy needs to be given more attention.

Time and location are two valuable pieces of information obtained from mobile phones that support many applications. However, the disclosure of data on these two attributes have been shown to leak privacy-sensitive information about the users, including their home and workplace location, as well as their routines and habits (Shilton 2009). For example, frequent visits to hospitals may allow employers to infer the medical conditions of their employees. Similarly, attendance at political events may provide information about users' political views (Liu 2007). Without any protection mechanism, the disclosure of location information may lead to severe consequences, ranging from social to safety and security threats. Aside from this, sound samples, pictures and videos can also be captured from participants and thus, conclusions about the number and identities of users' social relations can be drawn.

Location-based social networks (LBSNs) are a type of social network in which geographic services and capabilities such as geo-coding and geo-tagging are used to enable additional social dynamics (Symeonidis, Ntempos, and Manolopoulos 2014). In the past few years, time and location data have been used in many LBSN applications for different purposes. Simultaneously, increasing usage of personal location data has drastically changed the way in which people regard their location privacy. While almost three quarters (74%) of adult smart-phone owners use their phones to obtain directions or other information based on their current location, their trust in the provided privacy is not that

high (Zickuhr 2013). This can be understood by the number of users concerned about location sharing privacy, as the sharing of their location could be abused to disclose more sensitive personal information, such as home addresses and user identities. According to the research data shown in Madden et al. (2013), 58% of all teens have downloaded applications to their cell phone or tablet computer but 51% of teen application users have avoided certain applications due to privacy concerns. Moreover, 46% of teen application users have turned off tracking features on their cell phone or in an application and 26% of teen application users have uninstalled an application because they were worried about the privacy of their information.

Therefore, while providing interesting features to users, it is necessary to have privacy protections in LBSN applications.

1.1 Privacy

There is no internationally accepted definition of "privacy", since it can mean many things to different individuals. The level of privacy a person requires varies for different individuals at different times, locations and even different categories of application. It is important to understand that privacy considerations examine the rights, values and interests of individuals. In Bartoli et al. (2012), privacy is defined as an entity that can be subdivided into four dimensions. The first dimension is **privacy of personal information**. Personal information is any information that can directly or indirectly identify an individual's physical, physiological, mental, economic, cultural, locational or social identity. To fully maintain the privacy of personal information, an individual should be able to control when, where, how, to whom and to what extent the personal information will be shared. The second dimension is called **privacy of person**, which indicates the right to control the integrity of one's own body. It covers such things as physical requirements, health problems and required medical devices. The third dimension is **privacy of personal behaviour**. Individuals have the right to prevent any knowledge of their activities and their choices from being shared with others. Finally, the last dimension represents **privacy of personal communications**. This refers to a person's right to communicate without undue surveillance, monitoring or censorship.

Mobile social network applications encourage individuals to share their interests and knowledge widely. In LBSN applications, personal location information is being shared and communication is happening at all times. The privacy concerns address the four dimensions of privacy mentioned above. In order to meet the requirement of *privacy of*

personal information, a user's location information should not be shared to any other individuals or third parties without the user being notified and/or agreeing to terms such as when, where, how, to whom and to what extent. To meet the privacy requirement of *personal communications* requires that no third parties should learn about users' location tracks, contact histories, intentions or to-do schedules. This can be achieved by not providing any sensitive personal information to application providers, but in doing so, this might affect the use of applications. To address the *privacy of person*, we hide location so that a malicious user cannot physically reach the potential victims to cause any harm. We also address the *privacy of personal behaviour*, as we completely hide users' location coordinates so that their daily routines cannot be tracked. Users also control where and when to share their activities.

1.2 Categories of LBSN Applications

Thanks to advances in positioning technology and fast growing mobile online services, LBSNs also referred to as geo-social networks (GSNs) have gained huge popularity. These applications are enriching the widely-used online social networks with location-based services. By exploiting awareness of users' locations or knowledge of their proximity to points of interest, these applications are providing more fine-grained and personalized services to their users. Besides allowing people to socialize with others who share the same interests, many popular LBSN applications also use the geographical location to provide localized mapping of users. According to this major functions, we categorize location-based social networks into *dating applications*, *check-in applications* and *proximity notification applications*. We introduce these three categories of applications in the following subsections.

1.2.1 Dating Applications

Dating applications are location-aware mobile applications that provide a personal introduction system whereby individuals can find and contact each other over the internet to arrange a date, usually with the objective of developing a personal, romantic and/or sexual relationship. Users of a dating application service would usually provide personal information, such as age range, gender, interests and location, to enable them to search the service provider's database for other individuals.

Recently, dating applications have gained popularity. According to new research figures from Polakis et al. (2015), many popular dating applications, such as Skout, MeetMe, and Tinder, have gained between 10 to 50 million active users in the world. Taking Tinder as

an example: when a user logs in with a Facebook account, Tinder will upload the user's first name, photos, interests and age from the Facebook profile to create a Tinder profile. Tinder then finds the user's potential matches nearby (the user can narrow the matches down by searching by age and distance). If both users like each other, then they can start messaging.

To enable distance based matches, usually, dating applications require users to upload their current coordinates to their servers to calculate the distance between users. To address the privacy concern, many applications show an approximate distance to their users. For example, if the actual distance between two users is 630 metres, the server may show the distance as 1 km (Xu et al. 2015).

1.2.2 Check-in Applications

Check-in applications allow users to receive point-of-interest recommendations (like restaurants, gas stations, etc.) based on those users' current locations, or win vouchers and badges from retail services by sharing their locations to 'check-in'. For instance, Foursquare (Noulas et al. 2011) helps users keep up with friends, discover venues of interest nearby, save money and get rewards. SCVNGR (Li and Chen 2010) is another example, which builds a game "layer" on top of real-world places, by awarding discounts, badges and points to players for checking-in to places and completing challenges. A successful check-in is achieved by sending users' current coordinates to an application server to match the service provider's coordinates.

1.2.3 Proximity Notification Applications

Proximity notification applications allow users to receive notifications if their friends are within a certain range. This makes it easier for a user to catch up with friends in a big shopping mall, park or any other public place. For instance, Loopt (Humphreys 2007) automatically allows users to know when their friends are nearby. A proximity notification application can also be used to avoid seeing someone in an embarrassing scenario. For example, a man who is meeting his girlfriend for drinks, might like to receive notification if his ex-girlfriend is nearby, so that he could avoid an embarrassing situation of meeting both at some point.

1.3 Roles and Trust Assumptions

Location privacy requirements in LBSNs can be categorised into two groups, *mutual location proximity* and *one-way location proximity* (Hallgren, Ochoa, and Sabelfeld 2015). An example of mutual location proximity is discovering users in the vicinity (Šikšnys et al. 2009), without finding out the users' actual locations or distances. Dating applications and check-in applications fall under this category. One-way location proximity is of interest for the discovery of nearby people (e.g., doctors and police officers) without giving out the principal's location. Proximity notification applications could fit in both categories based on different scenarios. In LBSNs, there are three roles identified, which are classified as *initiators*, *servers* and *respondents*. Initiators are users who request location-based services. Servers provide the role of handling requests and exchanging information. Respondents are the users who give responses to the initial requests. Due to the different utility of applications, we will discuss the trust assumptions of each role in different types of LBSNs applications.

In the following description, we assume encryption to be in place for published content and traffic between members of the network. We further assume that the location services rely on the content encryption for access control and do not perform authentication themselves so that the ciphertext objects are available to all members of the network. Servers are only allowed to keep users' inputs and results for a short period of time.

In dating applications, initiators and respondents are both interested in knowing if they are close to each other. Since it is not one-way location proximity, an initiator is also a respondent and vice versa. Hence, either initiators or respondents could be dishonest users who provide fake inputs, or initiators/respondents could be malicious users who are attempting to track other users' locations. Servers are presumed to be honest but curious, and are trusted to compute the application functions based on users' inputs but might sell users' information to third parties.

In check-in applications, initiators request a location check to obtain points for rewards or to gain information. Respondents are usually service providers who give responses to initiators' proximity requests or forward information. Servers are assumed to be honest but curious, and are trusted to follow a protocol based on users' inputs but may log all messages and attempt to infer some further knowledge from the data received.

In proximity notification applications, social relationships between users can be asymmetric. In which case, initiators want to know if there are respondents nearby without the respondents knowing the result of the proximity check Alternatively they can be

symmetric, in which case initiators and respondents are mutual friends and agree to share the result of the proximity check. Servers are trusted to compute the proximity based on users' inputs but might sell users' information to third parties.

1.4 Adversary Model

Based on our analysis of related work on LBSNs, the architecture of LBSNs can be categoried into centralized and decentralized services. In current LBSNs, the central provider obtains all the data from users. An honest but curious application server could reveal users' privacy if the server is compromised by an adversary. Decentralized services often employ peir to peir networks to build communication between two peers. However, access right management, retrieval, and other administrative tasks of the service may be delegated to the LBSN users themselves. This entails that the members of a network are put in the position to abuse these roles. In this case, both random sniffers and friends have additional capabilities when compared to a centralized service (Greschbach and Buchegger 2012). Therefore, the user's location privacy can be compromised by at least two threats, an *honest but curious application server* or a *malicious user*. In the following we discuss different properties of adversary models, namely attractive attack targets, available information sources, possible inferences and impacts of attacks.

1.4.1 Attack Targets

Privacy preservation concerns both *user data content* and *information about that data*. We can categorize user data content into 1) user content and 2) metadata. *User content* includes profile information, posts (text, picture, video, link), comments on posts, liking posts, status updates and locations. *Metadata* compiles and summarizes basic information from user data, which can make finding and working with larger instances of data easier. For example, longitude and latitude that describe a specific location coordinate. Information about that data can be subdivided into a) social relation information and b) behavioural data. *Social relation information* captures users' relevant social connection graphs. *Behavioural data* captures the usage patterns of the service, which can expose users' preferences or routines.

All four categories contain pieces of information that are attractive targets for privacy invading attacks. This critical information can be used against a user in various ways.

1.4.2 Information sources

In a centralized LBSN, attack vectors are only available to the central provider which is presumed to be an honest and curious application server that we mentioned earlier. In a decentralized LBSN, an adversary might get information by observing network communications, sniffing metadata, or decrypting ciphertext by using background knowledge.

1.4.3 Possible inferences

An adversary will try to infer as much knowledge as possible from the collected information against the attack target. Collected information and background knowledge can be combined to interpret existing data or generate new knowledge. For instance, by observing the target IP address, with additional background knowledge about likely whereabouts (work place, home, friends, a favourite shop) the precise geographic location of the user can be inferred with high probability.

1.4.4 Attacks impacts

The adversary can be any entity interested in determining a user's location. We assume that an application server will follow its specifications and present no threat for its users. However, an adversary may compromise the server and potentially lead to disclosure of a user's location information. For instance, a government or law enforcement agency might request somebody's profile and location from the server. A malicious user could be a stalker, rapist, robber, liar or even a serial killer who has high technical skills. A user's location track could disclose his/her life routine and put the user's life in danger. A malicious user might fake his/her location to keep tracking the target's location until that he/she finds an opportunity to carry out a crime. A malicious user could even apply multiple fake profiles to perform a sybil attack to disguise his actual location (Piro, Shields, and Levine 2006). Our research aims to address these location threats and provide additional privacy protection.

1.5 Privacy Problems in LBSNs

To provide location privacy protection, many dating applications only give an obfuscated distance to the user, while several others prefer to provide quantifiable results. However, since the obfuscated location information is generated by the service providers themselves, these obfuscation solutions do not provide any privacy protection against curious applica-

tion servers. Moreover, an attacker can easily bypass the fuzziness of the results provided, resulting in the full disclosure of a potential victim's location, whenever the user is connected (Qin, Patsakis, and Bouroche 2014). Due to the disclosure of the location, the victim's safety and private information will be under threat by malicious users.

In order to achieve check-in functions, existing applications normally require users to share their location coordinates and social identities with the server. However, if the server is compromised, the users' home and work addresses, personal schedules and interests could be revealed. Therefore, there is a requirement for additional mechanisms to protect users' location privacy while using certain applications.

In order to achieve proximity notification, many existing mechanisms split the geographical area into cells and then compare whether two parties are located in the same cell. There are two major drawbacks for the existing mechanisms: 1) if there are only limited well-known objects in the cell, users' real locations will likely be disclosed to malicious users or third parties relatively easily; 2) two parties could be in locations where they are close to the edge of two neighbouring cells. However, those mechanisms will show that the two parties are not close to each other. Such circumstances are referred to as false negatives. In other instances, two parties could be in the same cell but far away on the different ends of a diagonal. Existing mechanisms erroneously show that the two parties are close to each other. These cases are referred to false positives.

1.6 Goal and Contributions

The goal of this thesis is to, 1 define the privacy problem in LBSN applications, and 2 design a solution so that users' locations cannot be revealed to malicious users and application servers while the application's main functions are being achieved. This requires the construction of a mechanism that can resist existing privacy attacks in location applications.

The contributions of this thesis are fourfold:

- We create a privacy-aware mechanism that fits all three categories of LBSN applications. Our approach enables a user or application server to check the location proximity result without knowing the user's actual location. It can be used in dating applications, check-in applications and proximity notification applications.
- We are the first to adopt cell tower identifier sets as a more deployable method for Location Tags. Such a method provides a dynamic shape of obfuscated region which

avoidentifiers miscalculating proximity as in other grid-based testing approaches. Moreover, our mechanism of using cell tower identifier sets, provides stronger resistance to privacy attacks than spatial cloaking. It is also easier to deploy than cryptography-based location tag approaches.

- We provide a higher level of privacy protection as we achieve the LBSN functions without requiring users' real location coordinates. Moreover, our approach protects a user's location privacy against both malicious users and application servers.

- We not only provide efficient proximity notification, but also predict proximity distance by analysing the similarity of data sets to give a better service to various social network applications. We devised a K-combination approach which is an accuracy enhancement to the k-shingling approach. Our approach takes every similar element between two data sets into account when comparing the similarity of those data sets.

1.7 Scope

This thesis focuses on solving privacy issues relating to LBSN applications. Most of the existing location privacy solutions try to obfuscate real locations with different methodologies. However, since these methodologies are based on the original location coordinates, the real location can be retrieved when these location privacy solutions come under specific privacy attacks. Location tags use environmental radio packages to define location instead of location coordinates, which is a safer approach to protect location privacy from attacks. Unfortunately, it is not practical to retrieve environmental radio packets using special sniffing tools on mobile devices. We have developed a solution to combat existing location privacy attacks by using cell tower identifier sets. In this work, location coordinates are not shared with any parities so no precise location data can be revealed by existing attacks. We also find that distance can be practically estimated by comparing the similarity of cell tower identifier sets, which enables us to achieve the proximity functions that LBSN applications intend to provide.

Nevertheless, by increasing privacy protection, a minor level of accuracy of the distance estimations is conceded. Unfortunately narrowing the accuracy of estimating distances is beyond the scope of this thesis, however, it could be further explored in future studies. Nonetheless, we find that our solution has met the basic proximity function and requirement for the aforementioned LBSN applications.

1.8 The Thesis Structure

The rest of this thesis is organized as follows. In the next chapter, ased on our study of the location privacy problems, we put forward our research questions and requirements. In chapter 3, we study state-of-the-art location privacy solutions proposed in the research literature with a discussion of their strengths and short-comings, and then provide further analysis and detail the privacy attacks on geo-social networks. After that, we propose our system model as a means of addressing the design challenges in this thesis. The following chapter illustrates experiment results of the correlation between distance and similarity from each of two test data sets and shows the accuracy of estimating distances that can be achieved by using cell tower identifier sets. In the final chapter we discuss our findings in the conclusion and outline possible avenues for future research specific to LBSNs location privacy and user security.

Chapter 2

Research Questions and Requirements

The location privacy test results, in Qin, Patsakis, and Bouroche (2014), show how users' locations can be disclosed by using a trilateration attack with fake locations with various degrees of approximated accuracy, despite obfuscation attempts. Often, users can benefit from being at a given location by receiving rewards. Hence, another problem is that malicious users could possibly manipulate the smart phone's GPS-based location by jail-breaking the phone. Malicious users could then benefit by falsifying their locations. Therefore, it is important to provide location privacy against malicious users and application servers.

2.1 Research Questions

In order to provide proximity-based services, personal location information is being shared through the internet for different LBSN applications. Hence, the risk of violating the key areas of privacy as mentioned above can occur.*Privacy of personal information* could be violated when location information is required to be shared with application servers and other users. While availing of application services, if users' locations are revealed to malicious users, for examplestalkers, their *privacy of person* would be violated. A curious employer, for example, might track their employees based on their location information, which could violate those users' *privacy of personal behaviour*. When the application server stores all users' locations, it could infer aspects of users' daily routines. For instance, Google has a location record including all the places where a user has been. Thus, users' daily routines are actually under the server's surveillance, which violates *privacy of*

personal communications.

A dating application user would usually like to know how far the other user is away from him/her so that they can assess their chance to meet. However, dating application users typically do not want to share their exact locations or distances with other users for safety reasons. In addition, to avoid their locations being tracked by application servers, users prefer to exchange obfuscated locations rather than exact coordinates. Moreover, the obfuscated locations should be able to hide users' real locations when under location privacy attacks. The duty of the application server is to verify if two users are in close proximity according to the given obfuscated locations. Users' profiles can then only be shared with nearby users as intended.

Check-in applications should allow users to show that they are within some range instead of the actual location to provide the requested information and receive an award. This will provide a level of privacy to users. Therefore, the application server would only need to know if the users are within a range from the check-in location but not the users' precise locations.

Friend-to-friend proximity notification applications allow users to receive notifications only when their friends are in geographical proximity to them. The alert is achieved by users agreeing to share location information in a friend-to-friend scenario. In this case, users agree to share their real-time locations to each other periodically. Regarding *privacy of communication*, users may not want to give their real-time locations to an application server, but may wish to give real-time obfuscated location updates. These obfuscated locations should allow the application server to correctly notify users if their friends are close. In addition, the obfuscated locations should be able to hide users' real locations when under location privacy attacks. The duty of the application server is to verify if two users are close to each other, based on the given obfuscated locations, so that a notification can be pushed to users if their friends are nearby.

From the point of view of privacy, users and servers do not need to know the actual location coordinates of other users while providing 'proximate' or 'nearby' services. A crucial challenge is to ensure that users' privacy is not violated by either servers or malicious users while achieving the main functionality of the application. The architecture must be applicable to a variety of location-based services and deployable on mobile devices as well.

Based on the discussion above, we postulate three main research questions. There are:

- In order to increase location privacy protection, is it possible to omit sending any actual location information to either users or application servers while still providing

social interactive functions such as 'people nearby' and 'proximity notification'?

- Can we create a solution which will satisfy the four dimensions of privacy detailed above?
- How can we make our approach deployable and scalable in mobile networks?

2.2 Research Requirements

The metrics used to measure the suitability of privacy solutions are based on the following three requirements.

1. The mechanism must have a strong resistance against privacy attacks. The mechanism that protects against the most privacy attacks is assumed to have the strongest resistance.
2. The mechanism should be able to protect users' location privacy against both malicious users and third-party servers. This metric requires that users' actual locations can not be detected by either application servers or malicious users.
3. The mechanism should function in various LBSN applications with a large number of users. The mechanism also needs to be mobile friendly (i.e., can be easily deployed on mobile devices).

Chapter 3

State-of-the-art Location Privacy Solutions

This chapter examines state-of-the-art solutions that have been proposed by other researchers in the past few years. We first discuss the advantages and disadvantages of each solution. Secondly, we examine which solution is suitable to be adopted in LBSNs. Finally in this chapter, we analyse and summarise the most suitable mechanisms to be used to protect location privacy in terms of resistance to location privacy attacks.

3.1 State-of-the-art Solutions for Participatory Sensing Applications

In Christin et al. (2011), the authors demonstrate how state-of-the-art privacy solutions can be applied to the different processing stages in a participatory sensing application system. The architectural components and their existing privacy countermeasures are illustrated in Figure 3.1.

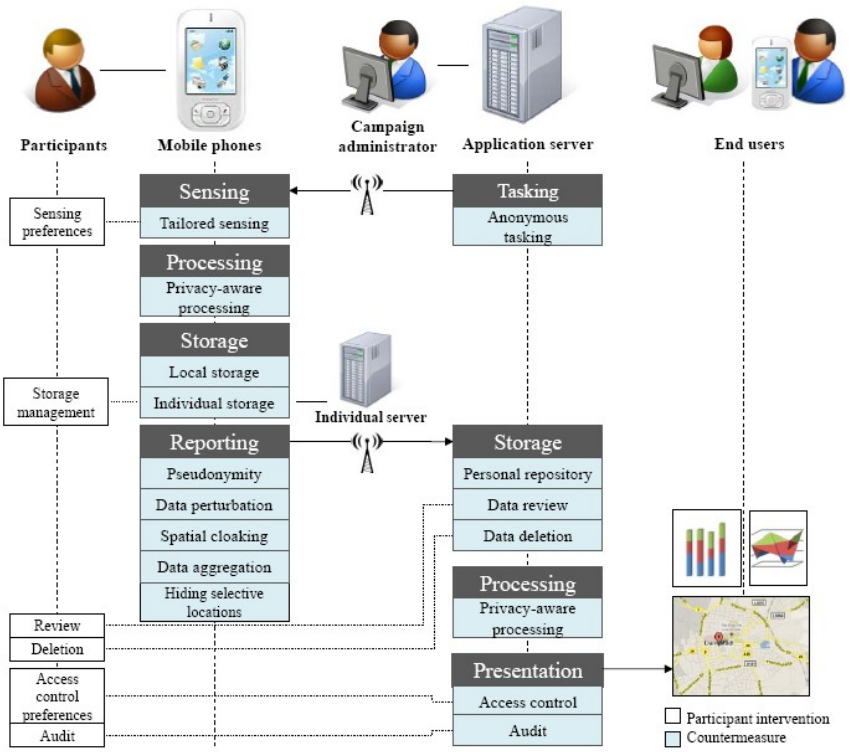


Figure 3.1: Countermeasures and their relationships to the architectural components (Christin et al. 2011)

Most existing participatory sensing applications consist of *participants*, *servers* and *end users*. *Participants* are the users who contribute to the sensing applications by gathering sensor readings using the mobile phones that they own and carry. *Servers* are usually maintained by application providers to collect and process the data. *End users* access and consult the data gathered by the participants according to their interests and preferences. This relationship is depicted in Figure 3.1 above. As we can see, participatory sensing on the participants' side could involve sensing, processing, storage and reporting. In the sensing component, a *tailored sensing* countermeasure can be used to control the data collection at the user level and allow the participants to express their privacy preferences by choosing when and what sensing data is to be shared. However, since participants will be able to selectively enable sensor measurements according to their personal conception of appropriateness, the quality of participatory sensing would be significantly influenced.

We focus on analysing countermeasures in the reporting component which supports the transmission of the sensor readings collected by the sensing component to the application server, using techniques such as pseudonymity, spatial cloaking, data perturbation, hiding sensitive locations and data aggregation.

Pseudonymity Instead of transmitting names in plain text, all interaction with the ap-

plication is performed under an alias. Pseudonym-based solutions provide anonymity and confidentiality to the user. An example of a pseudonymity solution is the anonymity-based TOR network, which is presented in Dingledine, Mathewson, and Syverson (2004). Before transmitting reports, the mobile phones select random relays along the path to the application server, instead of a direct route. The selected routes are then appended to the reports using a layered scheme, similar to onion layers. At each relay on the selected routes, a layer is removed using a symmetric key shared between this relay and the mobile phone. As a result, no relay knows the complete path from the report's source to the application server, but only the identities of preceding and following hops/relays. The participants feel more protected behind pseudonyms to share their readings without apprehension.

Drawback: This mechanism protects the participant's path when sending information to the application server. However, since the location is completely modified, this mechanism is not suitable for our target LBSN functions, such as 'people nearby' or 'check in' functions.

Data perturbation *Data perturbation* is about adding artificial noise, such as Gaussian noise, to disturb the sensor samples (Liu, Kargupta, and Ryan 2006). As the statistical characteristics of the noise model are known, the sum, average and distribution of the added noise over the data of all participants can be approximated. The community results can be estimated by subtracting the average noise time series from the sum of all individual perturbed datasets.

Drawback: Data perturbation relies on artificial noise. The result can be estimated if noise models are well known. Moreover, since the application server knows the noise models in advance in order to process data, it only protects privacy from malicious users but not application servers.

Spatial cloaking In *spatial cloaking*, the original value of the attribute is generalized by a value with less degree of detail. Spatial cloaking mechanisms include the well-known *k-anonymity* (Zhang and Huang 2009) and geographic obfuscation approaches (Ardagna et al. 2007). The key idea behind *k-anonymity* is mixing groups of k participants with a common attribute so that an adversary needs at least $k-1$ different pieces of background knowledge to eliminate the confusion to identify the real target. For example, the exact coordinates of the k participants are replaced by the name of the district of their current location. A classic geographic obfuscation ap-

proach is *cloaking granularity* including obfuscation and co-ordinate transformation approaches. Cloaking granularity uses different geometric obfuscation shapes considering some sensitive locations, or even provides mobile users with the ability to perform simple geometric operations (shifting, rotating) over their positions before sending them to the server. The idea of cloaking granularity is to protect the location privacy of the users by providing regions instead of precise positions. An advantage of spatial obfuscation approaches is that they can provide location privacy without a trusted third party (TTP) since users themselves can define the obfuscation area. However, this might affect the quality of service if clients are not provided with a precise user position.

Drawback: A risk to k-anonymity is the possibility of homogeneity attacks (see section 3.4) if the groups of k-participants are not well distributed. Furthermore, these approaches rely on a trusted third-party managing the generalization or perturbation of the locations for all participants. To generate the cloaked values, the participants need to report their exact locations to the third-party entity. A classic geographic obfuscation approach could suffer from a location-dependent attack. If an attacker (e.g. the service provider) can collect the historical cloaked regions of a user as well as the mobility pattern (e.g. speed limit), the location privacy of the user might be compromised based on the maximum movement boundary.

Data aggregation *Data aggregation* (Shi et al. 2010) does not rely on a central entity to protect data privacy, but on mutual protection within participants. Before transmitting data to the server, the mobile phones partially distribute their data among their neighbours. The mobile phones then upload the sensed data coming from their neighbours and the remaining of their own data.

Drawback: This approach only ensures data privacy protection if the nodes and the server do not conspire to breach the privacy of potential targets. If a user's neighbours are malicious and conspire with the server, then the server can easily reconstruct the complete data set from the uploaded slices and associate it to the user.

Hiding sensitive locations *Hiding sensitive locations* (Terrovitis and Mamoulis 2008) allows users to pre-define their sensitive locations. When users approach a location that has been previously defined as sensitive, the application generates fictitious location traces which intentionally avoid the selected location.

Drawback: This mechanism can be used as a privacy policy to allow users hide sensitive locations that they do not want to share in advance. However, the more locations that have been hidden, the harder it is to achieve application functions.

The mechanisms described above have been proposed to provide protection against different privacy threats in various applications. Since all the participatory sensing applications employ GPS sensor data, location privacy becomes the most important privacy concern. Hence, our research aims to provide a strong location privacy protection mechanism.

3.2 State-of-the-art Solutions for Location-based Applications

For privacy in location-based services, most previous works have focused on privacy in location queries, i.e., a model in which users report their "encrypted" location data to a central database server to perform range or k-nearest-neighbour (kNN) queries (Talukder and Ahamed 2010; Ghinita et al. 2008; Chang, Wu, and Tan 2011). The two most commonly used privacy solutions are location k-anonymity which is proposed to protect the user from identity disclosure (Khuong Vu and Gao 2012) and cloaking granularity that is used to prevent location disclosure (Li et al. 2008). However, in many mobile LBSN applications, the users not only query local information but also, personal information of other users and who is nearby. Thus, more location privacy protection needs to be addressed. In addition to k-anonymity and cloaking granularity principles, position dummies, mix zones, position sharing and cryptography-based approaches are some other existing options to be used to protect location privacy.

Position dummies The concept behind *position dummies* is that a user sends multiple false positions to the server together with the true position instead of just the true position alone (Kido, Yanagisawa, and Satoh 2005).

Drawback: One challenge of this approach is to create dummies that cannot be distinguished from the true user position. Otherwise, privacy protection is defeated by context linking attacks. For example, a malicious user could use a map to match these positions. If the dummy positions are located in some unreachable locations, the user's actual position can be easily distinguished. Moreover, this mechanism does not fit 'people nearby' or 'check in' functions if the dummy location is too far away from the actual location.

Mix zones *Mix zones*, proposed by Beresford and Stajano (2004), provide strong privacy by not sending any position updates within a defined zone. Users' identities are mixed together in the zone by changing pseudonyms to protect their identities. However, location privacy protection is only applied in the defined zone and user's entry and exit points can still be traced.

Drawback: Mix zones are designed to protect against disclosing certain sensitive locations. They do not fit the proximity functions in social applications.

Position Sharing *Position sharing* approaches split up the obfuscated position information into so-called position shares and distribute them among a set of non-trusted location servers so that each server only has a position of limited precision. Through share combination algorithms, multiple shares can be fused into positions of higher precision (Dürr, Skvortsov, and Rothermel 2011).

Drawback: This provides good privacy for applications that have multiple servers. However, if a server gets all the shares, it can calculate the actual locations.

Cryptography-based approaches *Cryptography-based approaches* encrypt users' location data to protect their privacy. Among cryptography-based approaches, 'proximity test' is the most suitable to be used in LBSN applications. *Proximity test* is a model in which location-based matching is done only between users, while the users' locations remain private. The only information being shared is whether or not two users are within a certain range or, in the same geographic region. In Mascetti et al. (2009), a proximity detection scheme based on service provider filtering is proposed. Privacy protection is achieved by a user-chosen location representation that controls its granularity. However, the protocol leaks coarse-grained location information to the server. (Narayanan et al. 2011) proposed a synchronous private equality test in which the server is used only to forward messages between the two users, but not perform any computation. It significantly reduces the privacy threat relating to third parties. However, this protocol relies on a-priori shared secret keys between each pair of users, which severely limits its applicability and scalability. In terms of the communication and computation cost, using ElGamal cipher texts (Schnorr and Jakobsson 2000) is not ideal if the number of users increases. In order to reduce the computation and communication cost, Narayanan et al. (2011) also proposed a fast asynchronous private equality testing protocol with an oblivious server. This protocol employs AES computation rather than ElGamal encryption. Compared with the

previous synchronous protocol, it is at least 10 times faster and has a 100 times lower communication cost. However, the protocol involves lightweight cryptographic primitives, and hence suffers traditional performance and scalability problems. An active attacker may inject any of the passed messages and the involved parties would not realize the change in the message. Nevertheless, these attacks might be prevented by deploying some simple mechanisms such as time-stamping (Saldamli et al. 2013). Moreover, the initial trust establishment among unfamiliar users in large scale mobile social networks has been a challenging task. It is not scalable and efficient enough to handle one-to-many proximity tests as studied in Zheng et al. (2012). Another researcher also suggested that users should be able to control their privacy levels via levelled publishing (Siksnys et al. 2010). The protocol is based on keyed hashing, which suffers from the dictionary attack.

Drawback: A traditional cryptography-based approach requires a pre-shared key exchange. Hence, it does not apply to large numbers of users in social network applications that includes friends and strangers. Moreover, the computation cost is another issue that needs to be considered.

3.3 Location Tags

A *location tag* is a collection of characteristic features derived from the unique combination of time and location. In other words, it is an ephemeral key that can only be obtained at a given time and a given location (Lin and Kune 2012). Therefore, any unpredictable ephemeral phenomenon associated with a location can be considered as a location tag, such as WiFi packets, 3G/4G packets, etc. Normally, these location tags can be derived from various electro-magnetic signals present in the physical environment, and must have two key properties:

Reproducibility Two measurements at the same place and time yield tags that should match with high probability. These two measurements need not be equal as strings.

Unpredictability The tags cannot be produced by an adversary if he is not at the specific place at the time.

Since the location tags of the two parties need to match, spoofing the location is no longer possible, which prevents against online brute force attacks. However, location tags come with a disadvantage, which is that users no longer have control over the granularity

of proximity. For instance, with WiFi packets, the neighbourhood is defined by the range of the wireless network.

3.3.1 Constructing Location Tags

Now we discuss several possible sources of location tags and ways to extract those location tags.

WiFi:broadcast packets WiFi broadcast packets include the source and destination IP address, sequence numbers, and precise timing information. These sources of information all offer varying degrees of entropy and then leave a rich potential for extracting location tags. Hence, we can sniff the WiFi traffic and compare the similarity of the number of packets between two users, such as 'ARP','BROWSER','DHCP', and 'NBNS'. Since those packets already contain high degrees of entropy, we could say these two users are proximate if those packets match with high probability. A shortcoming of using WiFi packets for location tags is that both users need to use the same wireless network.

WiFi or Bluetooth:Access point identifiers Wireless access points usually have a combination of SSID and MAC Address. The MAC address is a 48-bit integer fixed by the manufacturer and unique to each device. Bluetooth identifiers are similar to wireless access point. However, the differences in hardware led to significant differences in the list of addresses measured according to the measurements in Narayanan et al. (2011). For example, different mobile devices might find different numbers of bluetooth identifiers. We therefore believe that it is not a good source for location tags.

GSM or LTE:page messages A cell network issues a page on the broadcast paging channel of the base station covering a specific Location Area Code (LAC) when it contacts a mobile device. Each mobile station is assigned a unique Temporary Mobile Subscriber Identity (TMSI) or an International Mobile Subscriber Identity (IMSI). Since the TMSIs in paging request messages are local to each base station, there will be a disjoint sets of TMSIs if two phones are connected to different base stations. Thus, we could use the GSM paging channel as sources of location tag.

Audio Audio might be useful in certain limited circumstances to extract location features. However, the location tag will be limited to a room or small area, such as a coffee

shop. In addition, an audio privacy problem might be involved. Therefore, we do not consider that is a good source for location tags.

Atmospheric gases Sensors for recording CO, NO_x and temperature can be plugged into cell phones to collect real-time data. This information is another potential source of location tags. However, from the entropy point of view, the atmospheric gases readings are not guaranteed to be unique in different areas. Thus, it might not be considered as a good source of location tags.

Therefore, WiFi, GSM or LTE packets are the more suitable location tags for LBSN applications. Atmospheric gases or other location tags can be considered for use in enhancing the similarity comparison.

3.3.2 Drawbacks

Environmental patterns, such as WiFi and LTE packets, offer the unique unpredictability and reproducibility features to protect against location facking attacks. However, the location tag mechanism does not provide enough deployablility on mobile devices. Since mobile devices do not have the permission to access physical and data link layers, it requires particular sniffing tools to obtain those environmental packets. According to the results from Zheng et al. (2012), it takes at least 20 seconds to sniff LTE packets and generate an accurate location tag. Even if this mechanism can be deployed on mobile devices in future, draining power resources will still be a big problem. Moreover, the accuracy of proximity testing that uses location tags also depends on the quality of the sniffing tool, let alone the expensive price of the sniffing tool.

3.4 Attacks on Location-based Social Networks

To design better privacy protection, it is very important to have a good understanding of privacy attacks.

Information about user location can be inferred from geo-social networks and be exploited in many malicious ways[1]. However, location awareness opens up the possibility for even more attacks. For instance, based on collected location data, the home and work locations of users or even their identities can be recovered (Krumm 2007; Gambs, Killijian, and Prado Cortez 2014; Golle and Partridge 2009). Wernke et al. (2014) provide a classification of location privacy attacks including single position attacks, context linking

[1] `http://www.pleaserobme.com/`

attacks, multiple position attacks, attacks combining context linking and multiple position attacks, and attacks based on compromising a trusted third party component as shown in Figure 3.2.

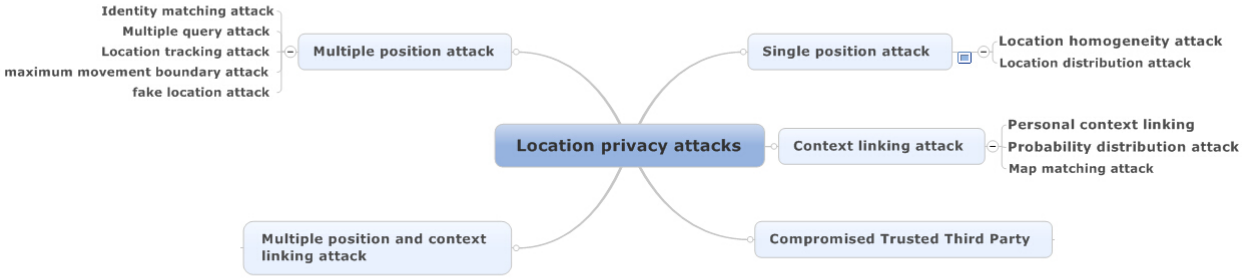


Figure 3.2: Location privacy attacks

A *location homogeneity attack* (Machanavajjhala et al. 2007) can be used against simple k-anonymity approaches. If k cluster members' positions are almost identical, the position information of each member is revealed. For example, if k neighbours are close to each other, then the target's location is identical (see Fig 3.3a). If k neighbours are distributed over a large area, then the position information is protected (see Fig 3.3b). An advanced location homogeneity attack can also utilize map knowledge to reduce the effective area size where users can be located. From Fig 3.3c, we can see that the protected user is more likely in the hospital after applying the map matching. Therefore, k neighbours in k-anonymity have to be diverse in a large area to avoid the homogeneity attack.

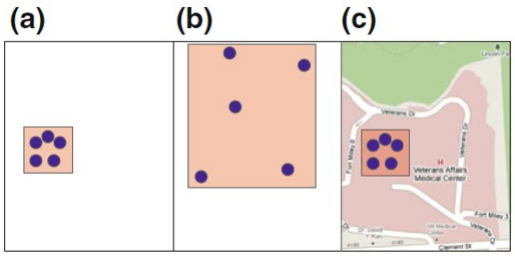


Figure 3.3: Location homogeneity attacks(Wernke et al. 2014)

A *location distribution attack* (Mokbel 2007) is based on the observation that k cluster

users are often not distributed homogeneously in space. In a k-anonymity approach, each user should be covered by the k cluster users. From the case shown in Fig 3.4, if $k = 4$ and A is the protected user, the calculated obfuscation area would be most likely the red area. However, the obfuscation area has to be extended to the dense area to cover other requested numbers of k users. If B is the protected user, the calculated obfuscation area would be the yellow area. In that case, user B's location would not be covered in such a dense area.

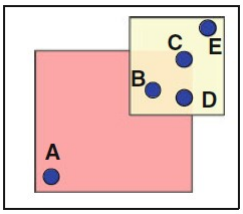


Figure 3.4: Location distribution attacks(Wernke et al. 2014)

A *context linking attack* (Machanavajjhala et al. 2007) is based on external background knowledge to decrease user privacy. The more context can be linked, the more privacy would be revealed. The context linking attack can be distinguished between three different kinds of attacks: *personal context linking attack*, *probability distribution attack*, and *map matching*.

A *personal context linking attack* (Gruteser and Grunwald 2003) is based on personal context knowledge about individual users, such as a pub that a user visits on a regular basis. If those user preferences or interests are known by an attacker, the attacker can decrease the obfuscation area to locations of pubs within the obfuscation area.

A *probability distribution attack* (Shokri et al. 2011) is based on gathered traffic statistics and environmental context information. If the probability is not uniformly distributed, an attacker can identify areas where the user is located with high probability.

Map matching (Krumm 2007) is based on a relevant map of the obfuscation area. An attacker can use semantic information provided by the map such as points of interest or type of buildings (bars, hospitals,or shops) to restrict the effective obfuscation area size.

The general idea of a multiple position attack is that an attacker tracks and correlates several position updates or queries of a user to decrease user privacy. *Identity matching* (Beresford and Stajano 2004) can be used to attack several pseudonyms based on equal or correlating attributes of the same identity. The *multiple query attack* and the *location tracking attack* are based on the analysis of several queries or location updates. The

attacker can correlate succeeding pseudonyms by linking spatial and temporal information of succeeding position updates or queries.

A *maximum movement boundary attack* (Ghinita et al. 2009) is based on the maximum movement boundary area where the user could have moved between two succeeding position updates or queries. As shown in Fig 3.5, based on the position of the first update performed at time T1, only a small part of the area of T2 is reachable within the maximum movement boundary.

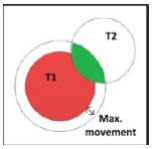


Figure 3.5: Maximum movement boundary attacks(Wernke et al. 2014)

Instead of using only one single attack, an attacker can also combine multiple position attacks and context linking attacks to undermine the user's location privacy. Moreover, a trusted third party (TTP) can be compromised by an attacker. Location information included in the query can be used as a quasi-identifier to re-identify the users by using snapshot location attacks (Gruteser, Grunwalddepartment, and Science 2003), query tracking attacks (Chow and Mokbel 2007), location-dependent attacks (Pan, Xu, and Meng 2012), trajectory attacks (Abul, Bonchi, and Nanni 2008; Gkoulalas-divanis, Verykios, and Mokbel 2009) and background knowledge attacks (Gedik and Liu 2005). In He, Liu, and Ren (2011), the location cheating attack has been discovered in which the attacker reports false locations to gain revenue by acquiring shopping coupons.

Some of the attacks mentioned above are similar to each other, such as the maximum movement boundary attacks and location-dependent attacks. Therefore, we summarize them into different attacks in figure 3.6. For a clearer presentation, we omitted the location distribution attack and the identity-matching attack, which are only applicable to k-anonymity and changing pseudonyms. We show the attacks as a matrix to measure the resistance of the presented approaches. If the mechanism can resist a certain attack, this is denoted by a $\surd$ in the main part of the figure.

	Location homogeneity attack	Personal context linking	Probability distribution attack	Map matching attack	Multiple query attack	Location tracking attack	Maximum movement boundary attack	Fake location attack	Compromised trusted third party
Pseudonymity		✓		✓	✓	✓	✓	✓	
Spatial Clocking	✓								✓
Data Perturbation	✓	✓		✓	✓	✓	✓		
Hiding sensitive locations	✓			✓			✓	✓	✓
Position sharing	✓				✓	✓	✓	✓	✓
Position dummies	✓			✓	✓	✓	✓	✓	
Mix zones	✓		✓	✓	✓	✓	✓	✓	✓
K-Anonymity			✓	✓	✓	✓	✓	✓	
Cryptography-based approaches	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figure 3.6: State-of-the-art approaches and their relationships to the general attacks

As we can see from the figure 3.6, cryptography-based approaches provide the strongest privacy protection against most privacy attacks. However, a traditional cryptography-based approach suffers the short-coming of lack of scalability. Hence, a new cryptography-based approach called location tag was introduced by Narayanan et al. (2011) and first studied by Qiu et al. (2009). When all the nearby users are in a similar environment, a location tag can be used as a unique public key without pre-sharing. Location tag methodology not only provides a cheating proof character but also offers a scalability feature.

3.5 Conclusions

In this chapter, in order to assess the applicability and effectiveness of location privacy approaches systematically, we studied the advantages and disadvantages of each of the state-of-the-art solutions. Most approaches to protect the user's identity against different attacks are based on k-anonymity. If the user wants to preserve location privacy without protecting his identity, the most popular technique to apply is spatial obfuscation. However, map matching as used against spatial obfuscation approaches has received great attention.

Most approaches that protect certain attributes, such as position and time, focus on single position updates and queries of a user but cannot resist a multiple query attack or a maximum movement boundary attack.

At present, only a few approaches (Solanas, Sebé, and Domingo-Ferrer 2008) can resist

a personal context linking attack. Most approaches cannot protect any combination of the attributes, such as identity, position, and time against such an attack.

Initially we introduced a classification of possible attacks that try to reveal the sensitive information. We then gave an overview of existing state-of-the art solutions to protect location privacy. Finally, after comparing these state-of-the-art solutions with the metrics we set in section 2.2, we detail a new table shown in Figure 3.7. From this table, it would appear that cryptography-based location tag approaches suit most of our requirements except deployability and efficiency. Spatial cloaking approaches fit the scalability and deployability but suffer from many privacy attacks. In order to increase a system's efficiency and deployability in acquiring location tags, we propose using surrounding cell tower identifiers, surrounding WiFi identifiers or environmental data instead of 3G/4G and WiFi packets. However, based on our study and practical tests, environmental data lacks stability and uniqueness. Hence, we believe that environmental data is not a good replacement for packet datasets. Due to the short range coverage of WiFi technology, users who have the similar surrounding WiFi identifier sets need to be in 100 metres proximity to each other. From the location privacy concern, such a small region will not be enough to protect users' location privacy. Therefore, using surrounding WiFi identifiers as a location tag is not a good replacement for packet datasets either. Thus, our research considers surrounding cell tower identifiers.

	Location homogeneity attack	Personal context linking	Probability distribution attack	Map matching attack	Multiple query attack	Location tracking attack	Maximum movement boundary attack	Fake location attack	Privacy resistance against Server	Suitable for social application	Scalable & Deployable
Pseudonymity		✓		✓	✓	✓	✓	✓			
Spatial Clocking	✓								✓	✓	✓
Data Perturbation	✓	✓		✓	✓	✓	✓			✓	
Hiding sensitive locations	✓			✓			✓	✓	✓		✓
Position sharing	✓				✓	✓	✓	✓	✓	✓	
Position dummies	✓			✓	✓	✓	✓	✓			
Mix zones	✓		✓	✓	✓	✓	✓	✓	✓		✓
K-Anonymity			✓	✓	✓	✓	✓	✓		✓	✓
Cryptography-based approaches	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

Figure 3.7: State-of-the-art approaches conclusion with our metrics

Chapter 4

Proposed System Model

In this chapter, based on our study of state-of-the-art location privacy solutions, we identify the problems with current solutions and propose an approach to the research gap.

4.1 Problems using Predefined Geographical Cells

As we discussed in the previous chapter, some proximity obfuscation solutions are grid-based, and use predefined geographical cells to obfuscate locations and compare proximity. Any coordinates located in the same cell will be presented as being 'close'. However, this suffers from false negatives and false positives. Consider Figure 4.1, which visualizes the worst case of an approach in which Alice's proximity is approximated by the gray cell; the approach considers Bob at the top right nearby, which is a false positive, but Bob at the bottom left far, which is a false negative. Even if Alice's position is in the center of such a cell, one can only exclude either false positives or false negatives, but not both. Since the cell is pre-defined, this solution also suffers from maximum movement boundary attacks.

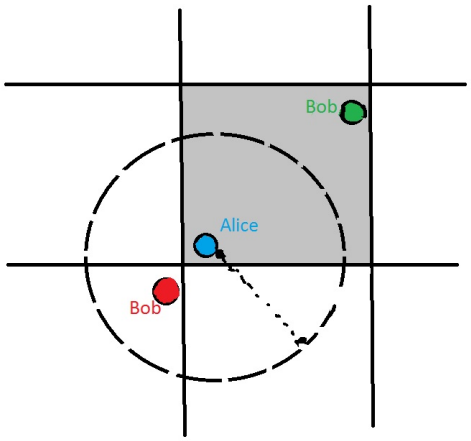


Figure 4.1: Grid-based testing

We could cut down the size of predefined cells to increase the accuracy of proximity estimation. However, to do so, will reduce the privacy protection as well. If the size of cells is too small, then users' locations might not be hidden to a sufficient level by the system. Finding well-defined geographical cells has been a critical problem. We propose that cells should be generated dynamically based on geographical factors and the distance between users. When two users are close to each other, the system should always be able to indicate that they are close and also be able to protect their location privacy. Cell towers are distributed in urban areas based on city planning and expected density of users. Moreover, each different cell tower has a different coverage range. Different combinations of cell towers can form different coverage shapes which increase the difficulty of executing a maximum movement boundary attack. A moving mobile user will be automatically connected to different cell towers that cover different regions. Therefore, using the coverage regions of surrounding cell towers as geographical cells can provide a dynamic grid for proximity testing. Two users who are close will always be in the same coverage. Thus, our proposal reduces the problem of false negatives and false positives.

4.1.1 Proximity Testing using Location Tags

Since Narayanan et al. (2011) proposed three protocols to address the underlying cryptographic problem of private proximity testing using location tags, many researchers have employed WiFi, GSM and LTE location tags to test proximity. Lin and Kune (2012)

demonstrated a very good example of extracting location tags from GSM networks. First of all, the mobile phones record all messages received on a special broadcast channel used by all phones in range of the same cell tower. Secondly, phones use a document de-duplication technique called shingling to produce a short string. The short string (sketch) represents the set of broadcast messages received. If two sets are similar, mobile phones will receive the same sketches with high probability. Otherwise, the sketches will be different. Finally, given a location sketch, phones can test for proximity using a private equality test on their sketches.

Zheng et al. (2012) also put forward a practical location tag construction method using 4G LTE networks and WiFi networks that provide location unforgeability. Their solution also allows users to tune their desired location privacy level and proximity range. This is a two-step protocol designed for one-to-many proximity testing between users that share no prior secrets, see Figure 4.2. Assume a user called Alice. During the first step, upon receiving the request from Alice, the server identifies a group of users designated by Alice and notifies users to construct their location tags simultaneously. A location tag in a 4G LTE network can be a control message that consists of temporary cell radio-network temporary identifier (TC-RNTI) and other information. Alice also embeds a temporary session key k in her location tag and sends it to the group to build a secure handshake and share the grid information later on. During the second step, users in the group first try to extract k. Only those within a coarse-grained proximity of Alice can succeed, which only occurs when sketches have a high similarity. The successful users then return a keyed hash of their current locations using a grid map representation to Alice for fine-grained matching. They employ a bloom filter to compactly represent the location tag while using the fuzzy extractor technique to accomplish a secure handshake.

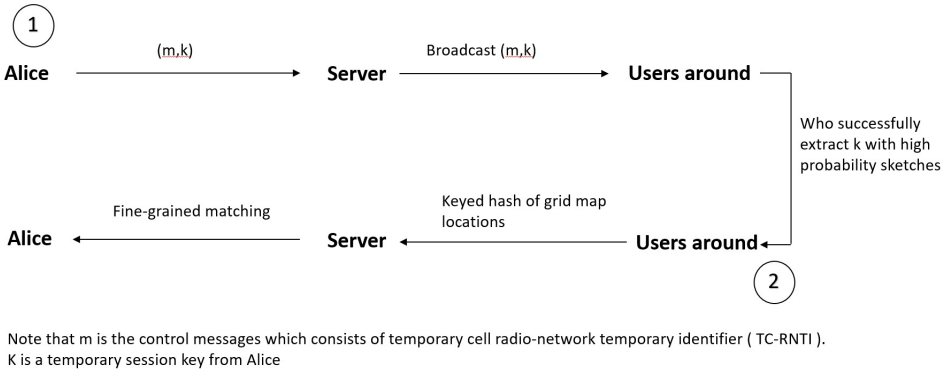


Figure 4.2: Processing flow

4.2 Introduction to Use Cell Tower Identifiers

In our system model, we assume that there is no trusted third party involved. Users do not give any identifiable location information to any users or servers. We cannot directly access cell packets from mobile devices. However, we can obtain the surrounding cell tower information directly from mobile devices. Noteworthy is that the surrounding cell tower information will change from time to time while the mobile user is moving. To the best of our knowledge, we are the first to employ surrounding cell tower identifiers as location tags to protect location privacy. Surrounding cell tower identifiers will be hashed into a sorted sketch which is compared for similarity. Neither malicious users nor servers can know the exact location information. If any party decrypted the hashed sketch, only a set of cell tower identifiers will be exposed, which is equal to a cloaking region. What is more, cell towers are placed in urban areas based on density of mobile users. It solves the problem of finding a dynamic geographical cell to test the proximity between two users since two nearby users will always be covered by a similar list of surrounding cell towers.

Assume that there are 5 users marked as A, B, C, D, and E. A, B and C are in close proximity to each other, but D and E are far away from A. As we can see from Figure 4.3, these five users are all covered by different sets of cell towers. Hence, they will obtain a different list of surrounding cell tower identifiers. Table 4.1 illustrates the list of surrounding cell towers which covers each of the users in our demonstration case. By comparing the list of cell tower identifiers, if each two sets of cell tower identifiers have

more intersections, two users will appear closer. Thus, it indicates that we can use the sets of cell tower identifiers as location tags to indicate relative proximity.

User	List of cell tower identifiers
A	106, 285, 258
B	258, 342
C	285, 258, 342
D	680, 511
E	198, 511

Table 4.1: List of cell tower identifiers for each user

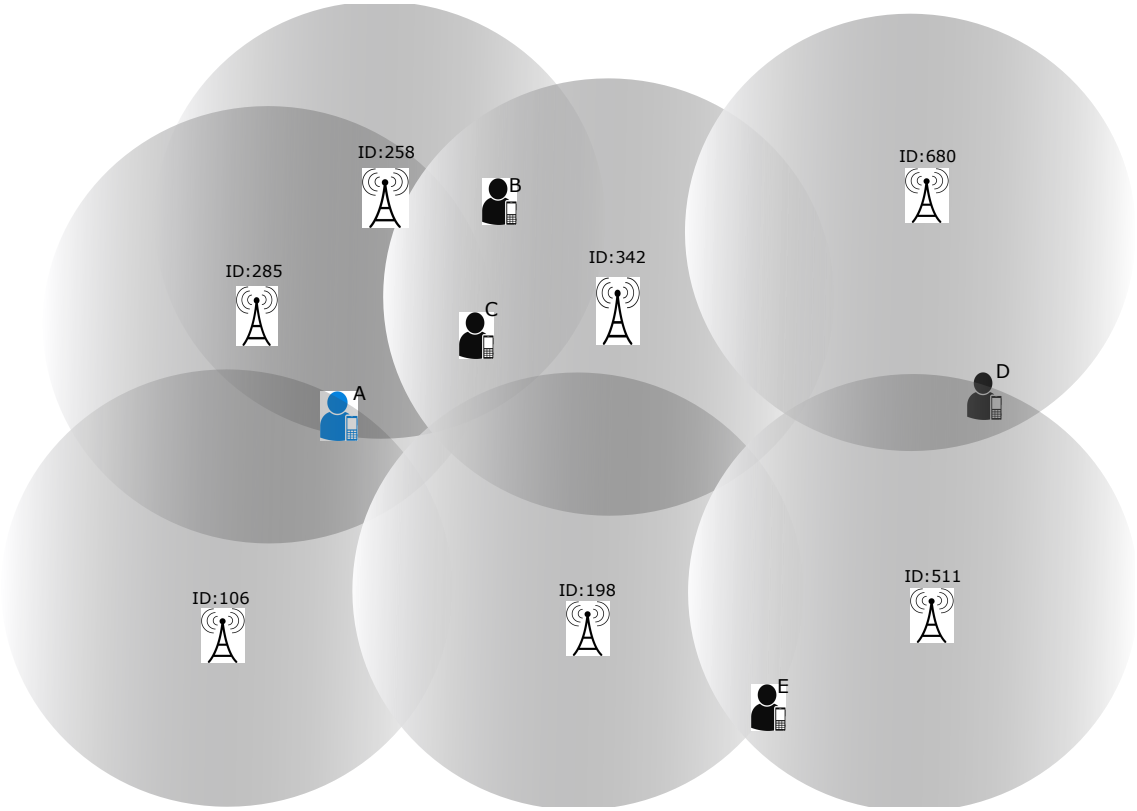


Figure 4.3: Case demonstration

4.3 Introduction to Communication Protocol

To use LBSN applications, all users have to register through the application server to create their profiles. After the registration, each user will have a unique private key S and a public key P. The private key S is used for building a handshake for private communication. Before two users make a private handshake, the system will run a proximity equality test based on the ambient dataset captured by users. Only the users who have similar datasets with high probability can successfully exchange private keys S. The public key P is used for delivering the dataset through the system in a secure way.

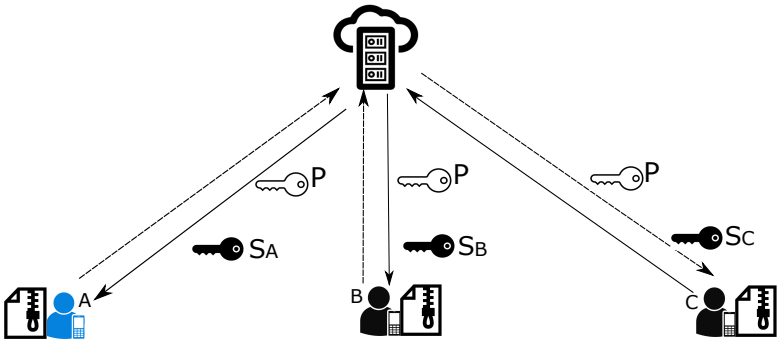


Figure 4.4: Registeration

We consider a sample of five users as an example, namely users A, B, C, D, and E. User A sends a query to the application server with his hashed cell tower dataset and time stamp. If user B, C, D or E would like to be seen by nearby users, they also send their cell tower data sets and time stamps individually to the server. Fig 4.5 details the communication exchanges from users A and B only, as those from other users would be identical. The server will compare A's dataset with the those whose time stamps are within a time interval from A's query. Here, we assume that only B and C have a similar dataset to A. In the end, A knows that B and C are near his location, but the server and other users know nothing about the location of A, B and C. This model can be easily

adopted to either mutual-way or one-way communication for different applications.

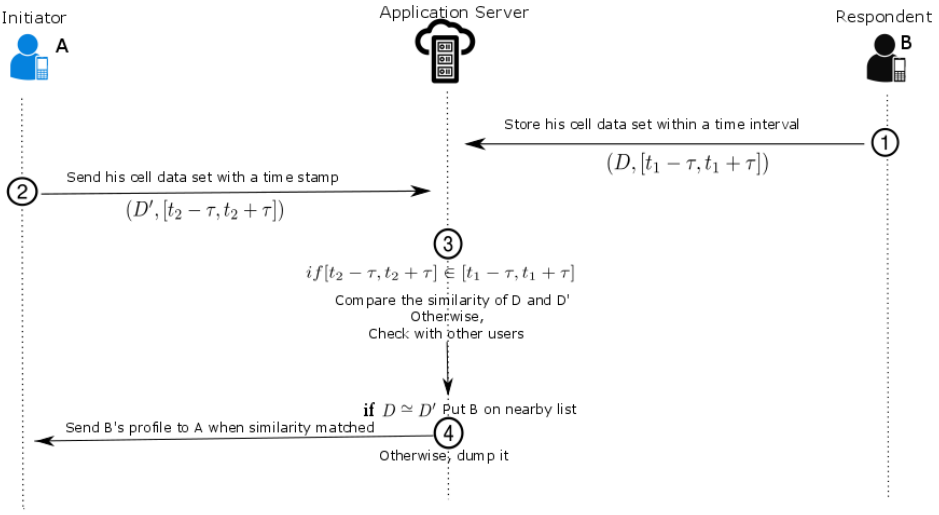


Figure 4.5: Communication protocol

4.4 Methodology for Hashing Cell Tower Identifier Sets

A potential shortcoming of employing neighbouring cell tower identifiers is that this mechanism lacks unpredictability since the identifiers of the cell towers and their locations are publicly known. Malicious users could fake their location by generating a fake sketch with a few neighbouring cell tower identifiers. Therefore, we add an encryption layer with a hashing function to hide the original dataset so as to mitigate this shortcoming. Once encryption is applied to the cell tower identifiers, without knowing the hash value, an adversary will not be able to fake/mimic/decode the correlated input data.

4.4.1 K-shingling

We adopt a mechanism called "Shingling" (Zheng et al. 2012) from the area of text fingerprinting. The shingling process is shown in Figure 4.6. A k-shingle is a k-tuple consisting of k consecutive elements of a set D, which is presented as a list of sorted elements. We define the k-shingling of a set D to be the set of all unique k-shingles of D, $S_D = S_1, S_2 \cdots, S_n$

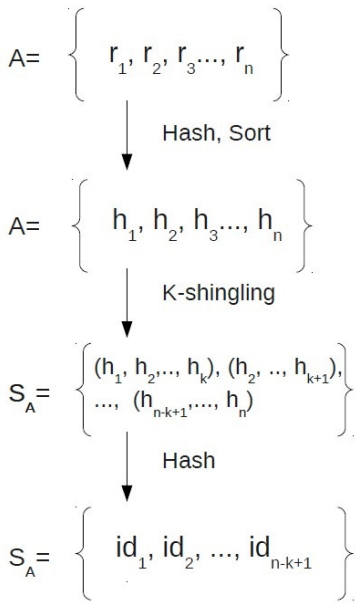


Figure 4.6: The shingling process

Nearly identical sets will generate nearly identical shingles. Each unique shingle can be indexed by a numerically unique id (UID). This will be done by hashing with a cryptographic hash. After shingling, set D will be converted into S_D which is a set of UIDs. By doing this we reduce the similarity test of cell tower identifiers to a similarity test of shinglings. The "resemblance" between sets A and B is defined by

$$R(A,B) = \frac{S_A \cap S_B}{S_A \cup S_B} \tag{4.1}$$

Example: We assume that there are two original data sets $A = 19, 8, 2, 25, 44, 33, 50$ and $B = 8, 19, 2, 25, 81, 44, 56, 72, 30$. A 3-shingling will be produced as following.

1. The first step is to hash and then sort these two data sets. When this process done, the original data sets will turn to $A = 2, 8, 19, 25, 33, 44, 50$ and $B = 2, 8, 19, 25, 30, 44, 56, 72, 81$

2. The second step is 3-shingling. A will produce
   $S_A = (2, 8, 19), (8, 19, 25), (19, 25, 33), (25, 33, 44), (33, 44, 50)$.
   B will produce
   $S_B = (2, 8, 19), (8, 19, 25), (19, 25, 30), (25, 30, 44), (30, 44, 56), (44, 56, 72), (56, 72, 81)$

3. Then we hash the data sets S_A and S_B by using the sum of 3 shingles. The final data sets are $S_A = 29, 52, 77, 102, 127$ and $S_B = 29, 52, 74, 99, 130, 172, 209$

By using the resemblance equation, the similarity of the two original data sets is roughly equal to 0.45. However, the similarity result from 3-shingling is 0.2. This shows that we have to sacrifice a certain level of accuracy to protect privacy. In order to increase the accuracy while protecting privacy, we could use 2-shingling instead of 3-shingling. However, doing so may increase the probability of a malicious user being able to identify the coding pattern and break/decode our encryption. It is important to see how 2-shingling and 3-shingling will affect our overall results of similarity versus distance. Aside from the loss of accuracy, there is another drawback when using k-shingling, due to the use of consecutive elements. For instance, if similar elements do not line up side by side, after applying 3-shingling, the similarity result will be zero.

4.4.2 K-combination

K-combination is a modified version of k-shingling, which produces a k-tuple consisting of k elements of the set D. These k elements are combinations of selected k hashed cell tower identifiers from the data set. We take all similar elements between two data sets into account when comparing the similarity of those data sets, so that there is no consecutive elements effect from the previous methodology. If there is a similar element between two data sets, the element will always be counted in our comparison. In addition, we do not need to sort the data sets at the first stage, which saves execution time. The K-combination process is illustrated in Figure 4.7.

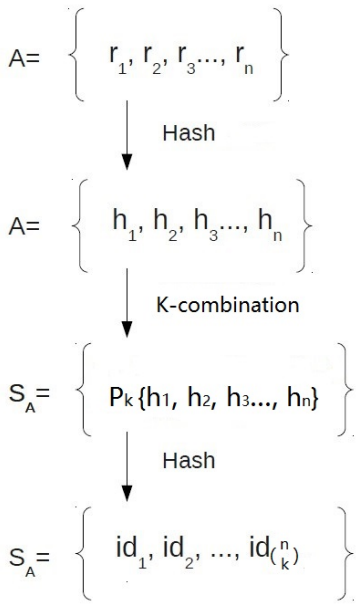


Figure 4.7: K-combination process

Example: We assume that there are two original data sets $A = 18, 8, 2, 25, 44$ and $B = 8, 19, 2, 25, 44$. We will do a 3-combination as follows.

1. The first step is to hash these two data sets. Since these two samples are all integers, so they will return to $A = 18, 8, 2, 25, 44$ and $B = 8, 19, 2, 25, 44$

2. The second step is 3-combination. A will produce
   $S_A = (18, 8, 2), (18, 8, 25), (18, 8, 44), (18, 2, 25), (18, 2, 44), (18, 25, 44), (8, 2, 25), (8, 2, 44), (2, 25, 44)$
   B will produce
   $S_B = (8, 19, 2), (8, 19, 25), (8, 19, 44), (8, 2, 25), (8, 2, 44), (19, 2, 25), (19, 2, 44), (19, 25, 44), (2, 25, 44)$

3. Then we hash the data sets S_A and S_B by using the sum of 3 combinations. The final data sets come as $S_A = 28, 51, 70, 45, 64, 87, 35, 54, 71$ and $S_B = 31, 52, 71, 45, 54, 46, 65, 88, 71$

By applying the resemblance equation, the similarity of the two original data set is equal to 0.66. The similarity result by applying 3-combination result is 0.215. In this case, the similarity result using 3-shingling is 0. This shows that the result of using our k-combination is much more accurate than k-shingling on this scenario. In other samples, the result of using k-combination would be even closer to the original similarity test result than using k-shingling. We study the effect of applying k-combination on the overall results of similarity versus distance in the next chapter.

Chapter 5

Evaluation and Results

This chapter initially presents an experiment to evaluate the correlation between distance and similarity of cell tower identifier datasets. The experiments indicate that correlation between distance and similarity is still valid after the original datasets have been hashed using our methodology. The results of the evaluation are discussed, and this chapter ends with a summary of the results and findings.

5.1 Correlation between Similarity of Cell Tower Identifier Sets and Distance

Cell towers are distributed around urban areas with individual identifiers for different mobile networks. Based on the usage of communication in an area, different cell towers also have different transmission power which provides different communication ranges. Any given location should be covered by a number of different cell towers based on their communication ranges. In other words, any given mobile user in a particular location should be covered by a set of nearby cell tower identifiers. If two mobile users A and B are close enough, then the two of them should be covered by sets of nearby cell tower identifiers with high similarity, which means that the intersection between the two sets covering A and B is large. Alternatively, if two mobile users are far away from each other, the intersection between the two sets covering A and B is small.

OpenCellID (Ulm, Widhalm, and Brändle 2015) is the world's largest community project that collects GPS positions of cell towers, used free of charge for a multitude of commercial and private purposes. The OpenCellID project was primarily created to serve as a data source for GSM localisation. More than 49,000 contributors have already registered with OpenCellID, contributing more than 1 million new measurements every

day on average to the OpenCellID database. By using the database from OpenCellID, it is possible to measure the correlation of similarity with distance using real-world data. Another purpose of this experiment is to test the maximum distance between two user locations that have the same set of results, which gives the maximum level of location privacy that we can achieve.

5.1.1 Experimental Methodology

Data Selection

We have manually selected three different areas corresponding to different densities of mobile users in Dublin. To do so, we first record the GPS longitude and latitude of the selected point, then add 0.05 to both longitude and latitude to create an area. After a test area is defined, we use an SQL tool to obtain all the Vodafone GSM cell tower identifiers in the selected area from the OpenCellID database. We map those cell towers' locations to X and Y coordinates relative to the original selected location. By repeating this process, the other two areas are created in the same way for testing.

Converting Longitude and Latitude to X and Y

The longitude to the east direction is represented as the X axis, and the latitude to the north direction is represented as the Y axis. In order to reduce errors from distance calculations, when mapping longitude and latitude to X and Y coordinates, we used Google maps as a reference to select the correct formulas. According to our test results, we adopted the spherical law of cosines formula as our methodology for mapping longitude to the X axis by setting $lat_1 = lat_2$. The formula is as follows:

$$X = \arccos(\sin(lat_1) * \sin(lat_2) + \cos(lat_1) * \cos(lat_2) * \cos(lon_2 - lon_1)) * R \qquad (5.1)$$

For mapping latitude to the Y axis, we implement the north-south distance between two lines of latitude:

$$Y = R * (lat_2 - lat_1) * pi/180 \qquad (5.2)$$

Note that R is the radius of the earth, which equals 6,371,000 metres.

Similarity Testing

For similarity testing, we notice that those points that are located close to the edges of the area can only receive a partial set of cell tower identifiers. Some other cell towers which are beyond the X and Y coordinates are not selected as part of our test area. In order to ensure that every single test point receives all the possible cell tower identifiers, we pick sample test points to be within one square kilometres area in the centre of distribution of all selected cell towers. The distance between each test point is 20 metres.

For each test point, we calculate the distance between each cell tower. Calculation follows the formula below:

$$D = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \tag{5.3}$$

If the distance is smaller than the given cell tower range, the test point will receive this cell tower ID. Thus, each test point will have a set of cell tower identifiers in its coverage area. Assuming that point A receives a set of cell tower identifiers S_A and point B receives a set of cell tower identifiers S_B, the similarity between this two points is

$$R(A, B) = \frac{S_A \cap S_B}{S_A \cup S_B} \tag{5.4}$$

By using the similarity and distance calculation formulas, we generate similarity versus distance for each pair of test points.

5.1.2 Experimental Results

In this section, we present our experimental plots and findings about the correlation from each test.

Distribution of Cell Towers

Figures 5.1, 5.2 and 5.3 individually indicate the cell tower distribution in our three different test areas. Each area was selected manually by choosing a range of longitude and latitude, and then converting to X and Y coordinates in metres. Table 5.1 shows the variables we used for those three tests.

Summary of Test Area				
	Longitude range	**Latitude range**	**City Location**	**Fixed parameter**
Test Area 1	-6.28 to -6.23	53.33 to 53.38	city centre	0.05
Test Area 2	-6.28 to -6.23	53.27 to 53.32	close to city centre	0.05
Test Area 3	-6.20 to -6.15	53.23 to 53.28	not close to city centre	0.05

Table 5.1: Summary of Test Areas

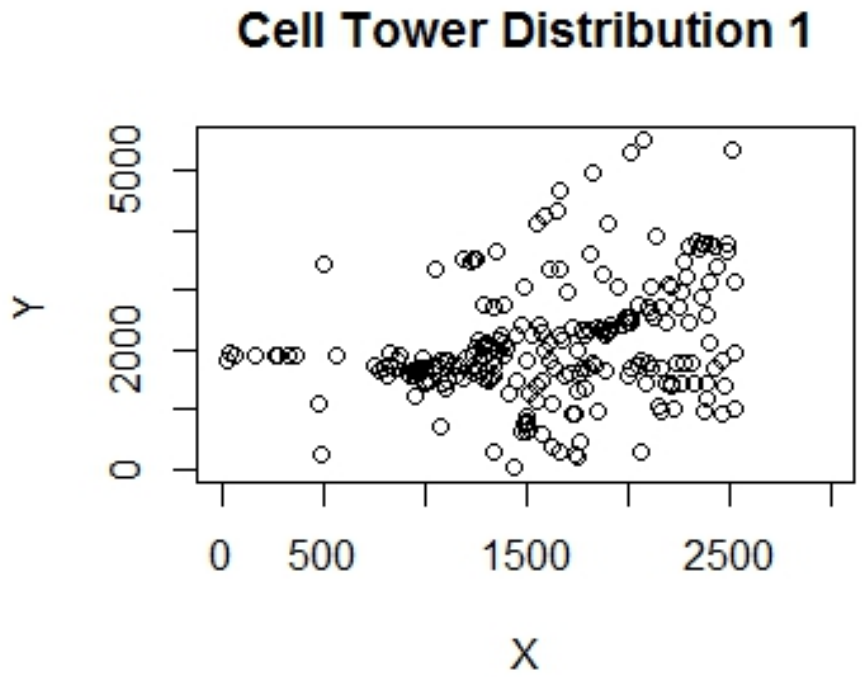


Figure 5.1: Cell Tower Distribution in Test Area 1

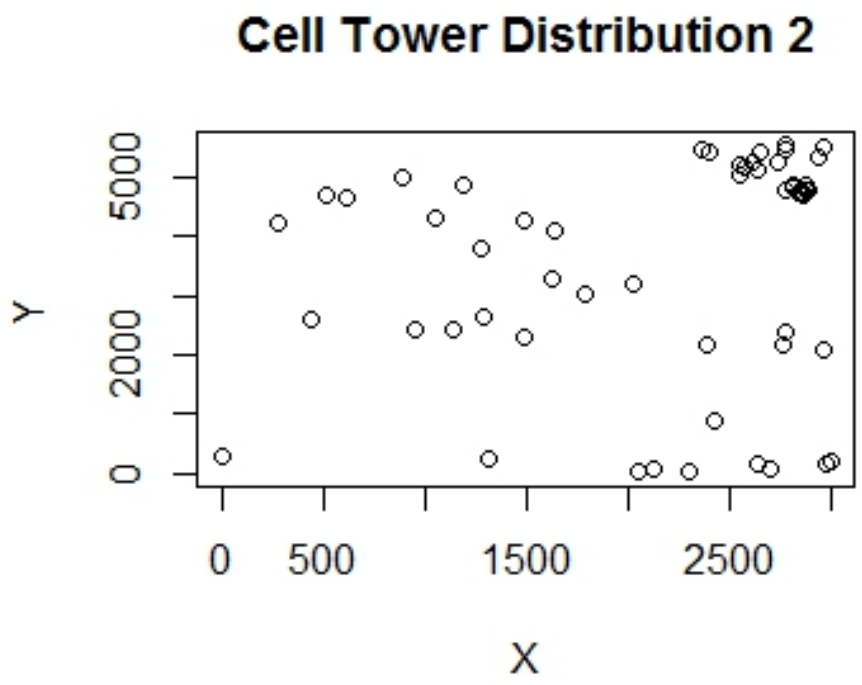


Figure 5.2: Cell Tower Distribution in Test Area 2

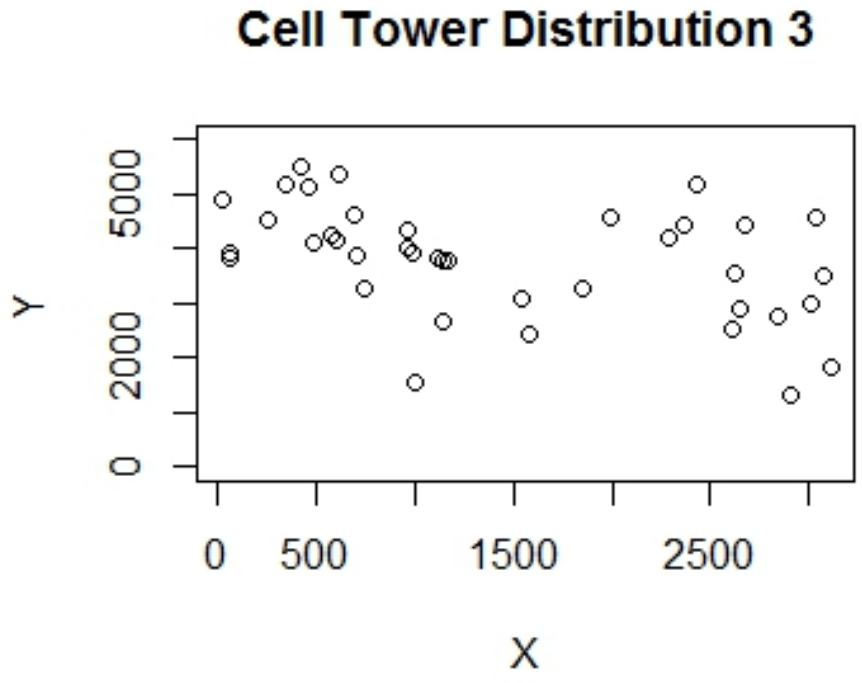


Figure 5.3: Cell Tower Distribution in Test Area 3

The results from Test 1 are measured in the central area of the city with a very high density of mobile users. The number of cell towers in Test 1 is larger and more dense than in the other two areas. From the three distribution plots, we can see that the number of cell towers is correlated with the density of mobile users. The number of cell towers decreases when the number of people reduces. From the OpenCellID database, we also find that the coverage range of the cell towers in the country side is larger than in the city centre.

Distance versus Similarity

Within the selected area, for each two points, we calculate the similarity of their received set of cell tower identifiers and the distance between them. The distance versus similarity plots for each test are shown in Figures 5.4, 5.5 and 5.6. The regression parametres in relation to distance vs similarity can be seen in Table 5.2.

Table 5.2: Regression Parameter for Each Test

	Correlation	Standard error	Intercept
Test Area 1	-0.8013584	863.7934	135.3
Test Area 2	-0.840547594	3010.306	131.5
Test Area 3	-0.6587015	1220.4803	186.4

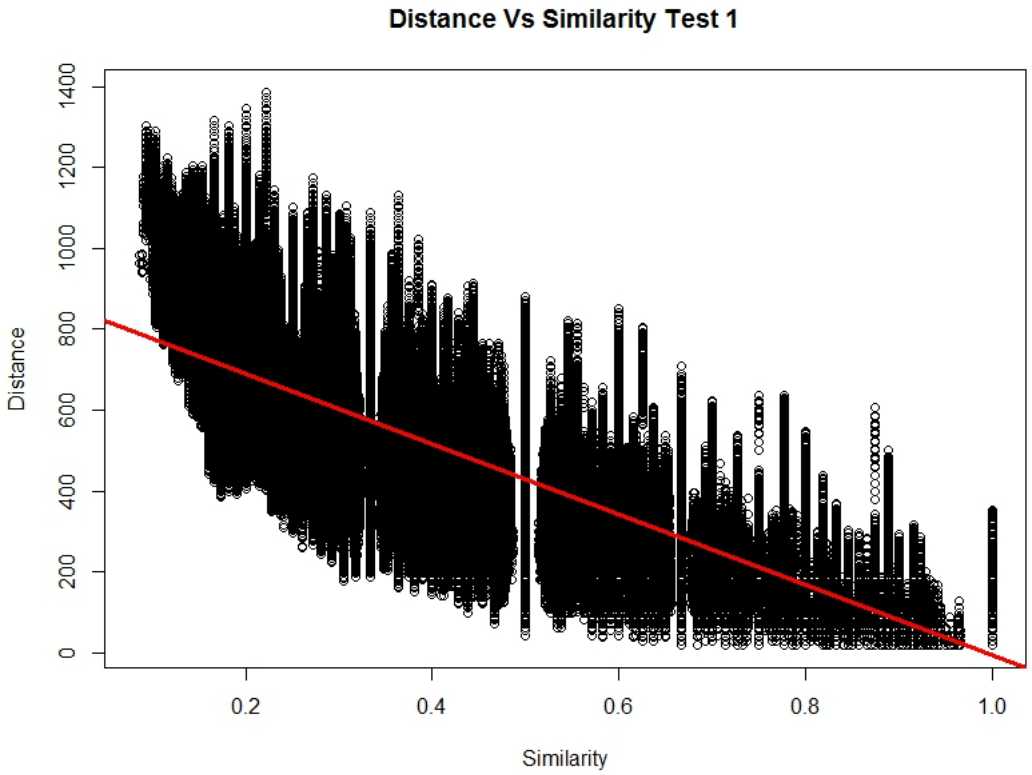


Figure 5.4: Distance vs Similarity in Test Area 1

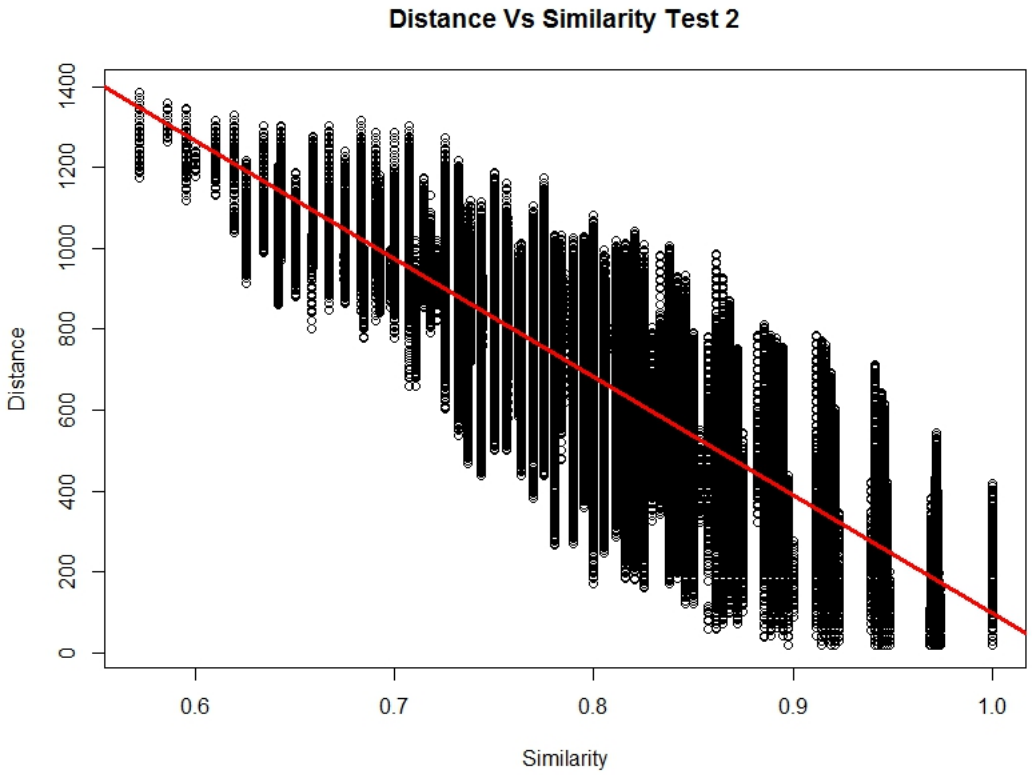


Figure 5.5: Distance vs Similarity in Test Area 2

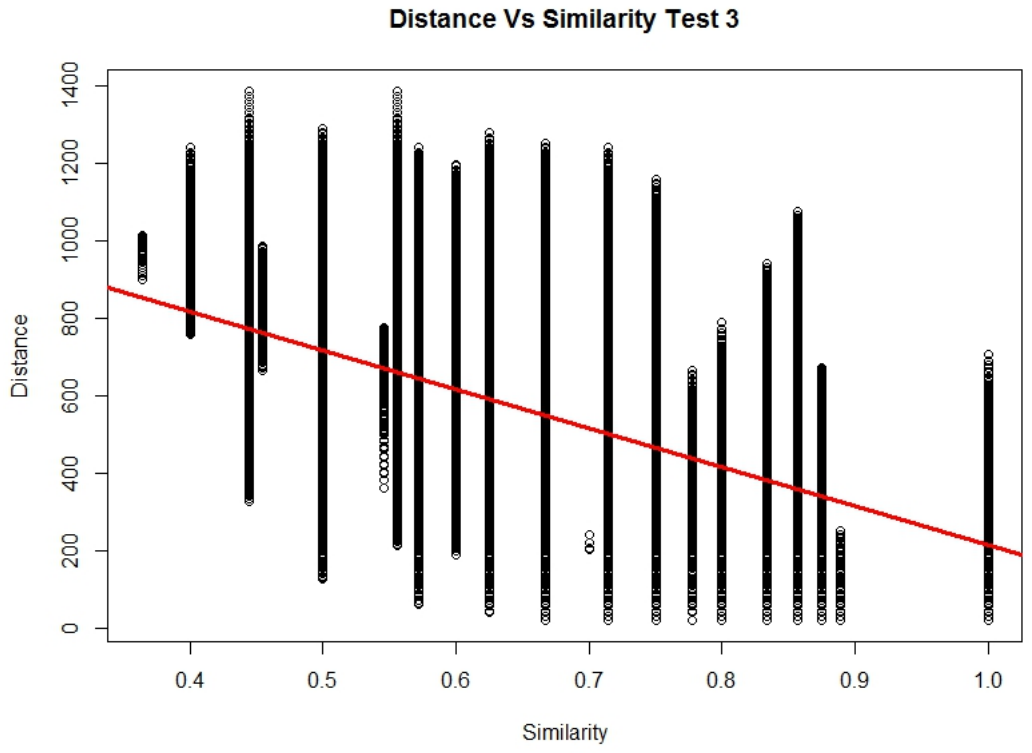


Figure 5.6: Distance vs Similarity in Test Area 3

Findings: According to the results shown, the distance is highly correlated to similarity when the distribution of cell towers is dense. The relationship between distance and similarity varies from location by location. However, the distance between two points can be estimated from the similarity value when the similarity of two sets of cell tower identifiers is smaller than 0.4. This means that, for privacy concerns, a malicious user cannot predict other users' actual distance or locations from a given similarity. Conversely, the system can predict whether two users are close or far from each other, which fits our requirement but does not disclose users' locations. Moreover, by monitoring the distance value when similarity equates to 1.0 in Figures 5.4, 5.5, and 5.6 , we can see that the distance range between two points with the same set of cell tower identifiers increases when the density of people reduces. This means that, by using cell tower identifiers as the attribute for predicting user's distance, the level of location privacy protection will automatically increase when the level of density of citizens reduces.

Checking Linear Regression Assumptions

To fit the linear regression, we made assumptions as follows:

1. The distance can be expressed as a linear function of similarity.
2. Variation of observations around the regression line is constant, known as homoscedasticity.
3. For a given value of similarity, distance is normally distributed.

To check if our results meet the above linear regression assumptions, we generated residuals plots for each test as shown in Figure 5.7, 5.8, and 5.9. The fitted values in those plots present the linear regression of distance with similarity. Residuals are the errors that presents the variation of observation around the regression line.

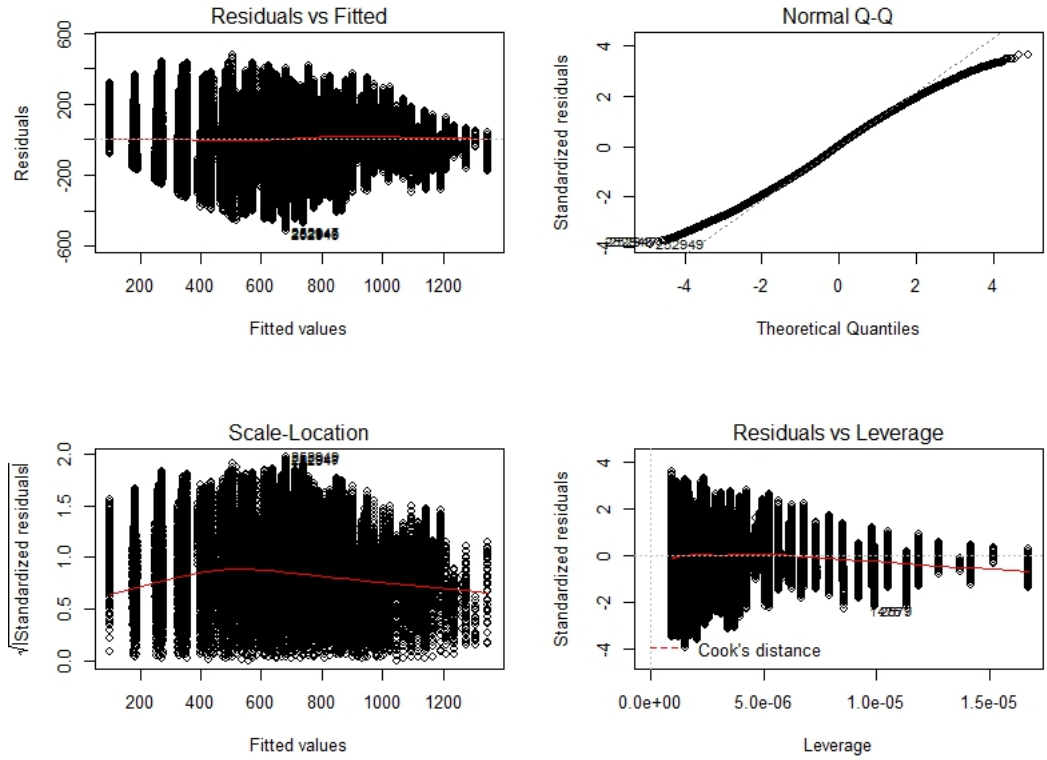


Figure 5.7: Residual Plots for Test Area 1

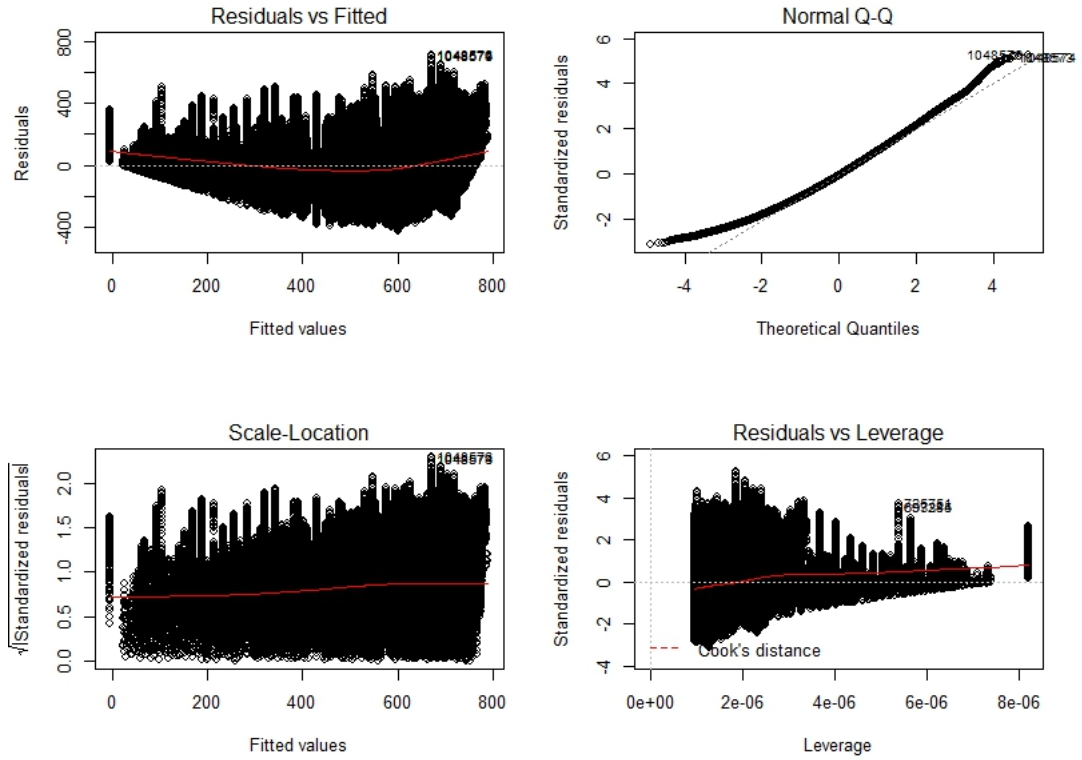


Figure 5.8: Residual Plots for Test Area 2

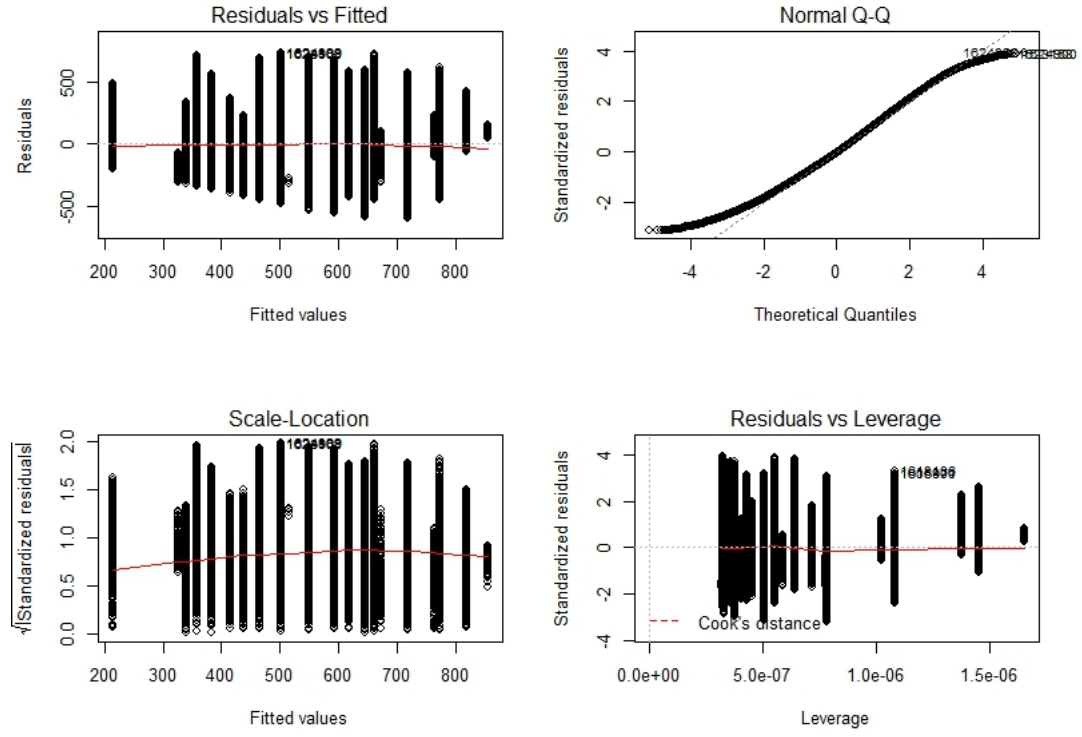


Figure 5.9: Residual Plots for Test Area 3

Findings: From the Normal Q-Q plots, we can see that the errors/residuals are not fully normally distributed but roughly normally distributed. The other three residual plots are used to indicate if the variation of observation around the regression line is constant. Since the red line is roughly flat, we can conclude that the errors/residuals approximately meet our linearity assumptions. However, the variation is slightly decreasing when the predicted value is getting large from Figure 5.7. And the variation is slightly increasing when the predicted value is getting large from Figure 5.9. These residual plots indicate that distance and similarity are fairly normally distributed, which meets the linear regression.

Similarity versus Distance

In order to see if similarity can be estimated by a given value of distance, we have also plotted similarity versus distance using the same experimental data. The similarity versus distance plots for each test are shown in Figures 5.10, 5.11 and 5.14.

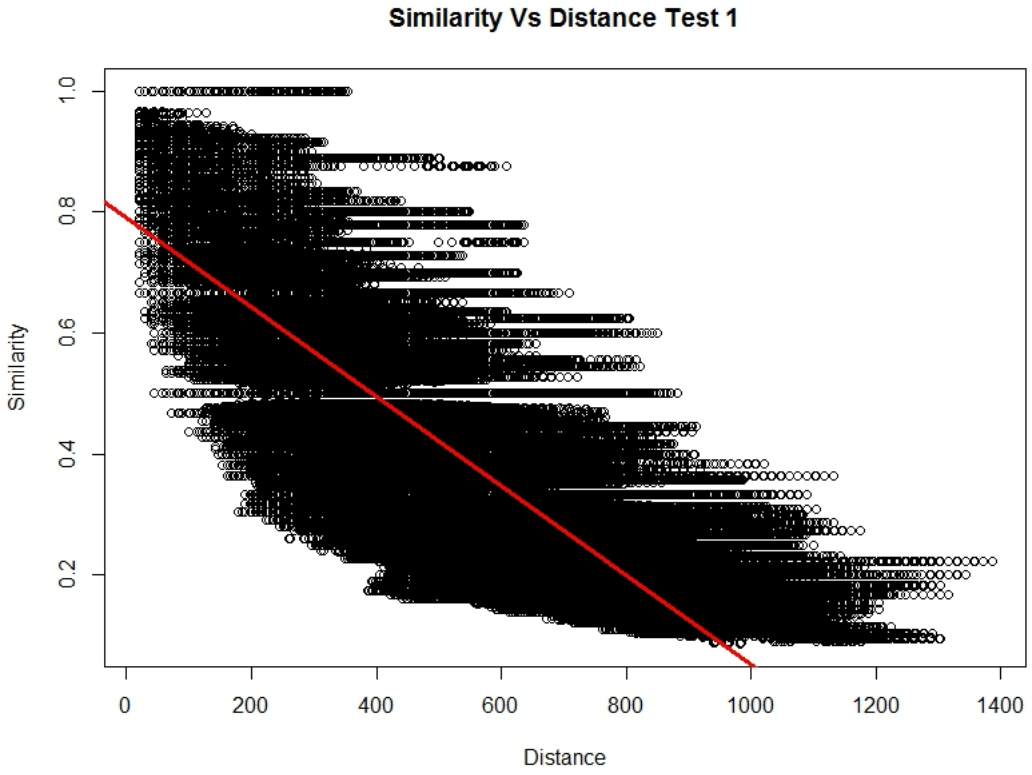


Figure 5.10: Similarity versus Distance in Test Area 1

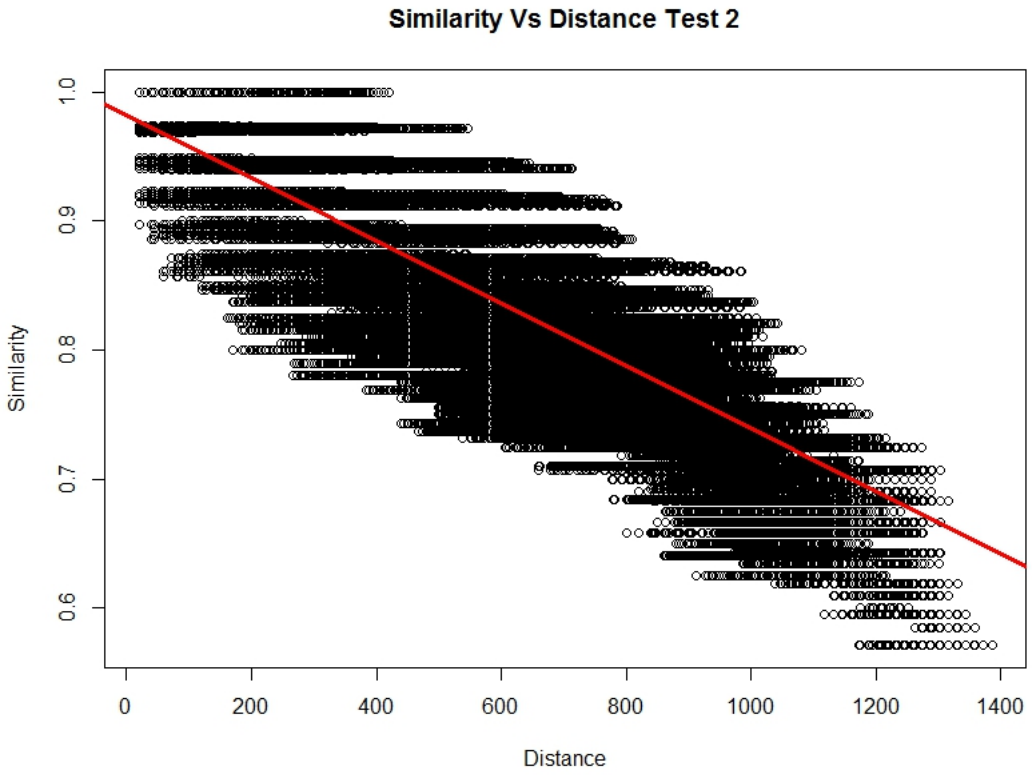


Figure 5.11: Similarity versus Distance in Test Area 2

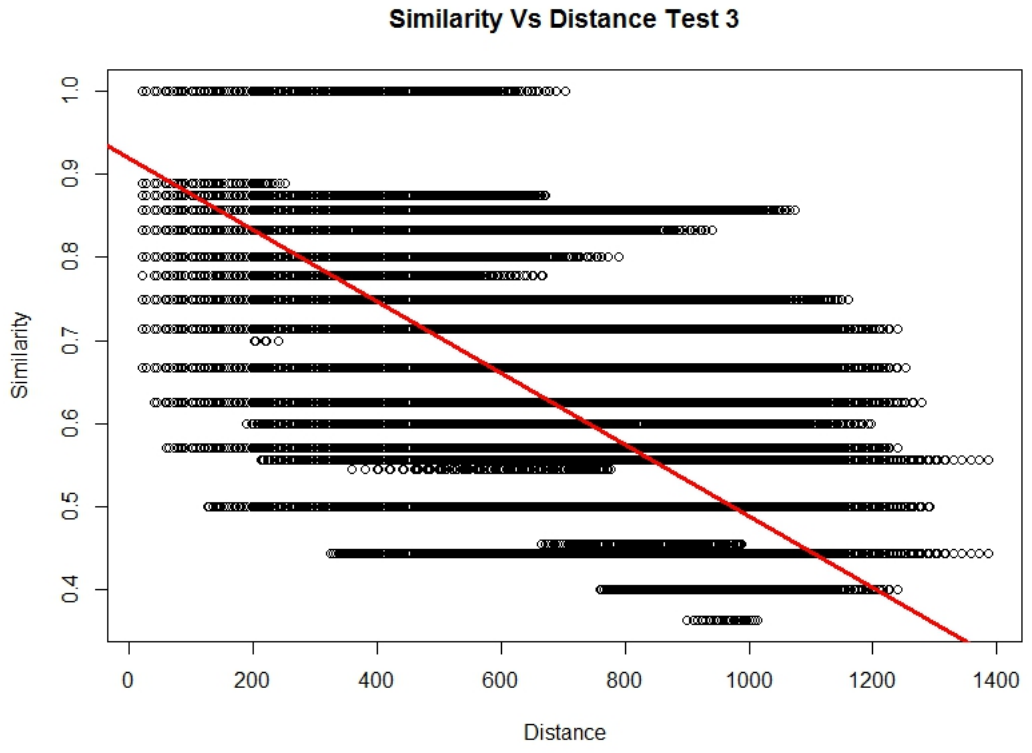


Figure 5.12: Similarity versus Distance in Test Area 3

We made similar linear regression assumptions as follows:

1. The similarity can be expressed as a linear function of distance.
2. Variation of observations around the regression line is constant, known as homoscedasticity.
3. For a given value of distance, similarity is normally distributed.

In order to check if these similarity versus distance plots meet the linear regression assumptions, we selected two tests and generate their residual plots as shown below.

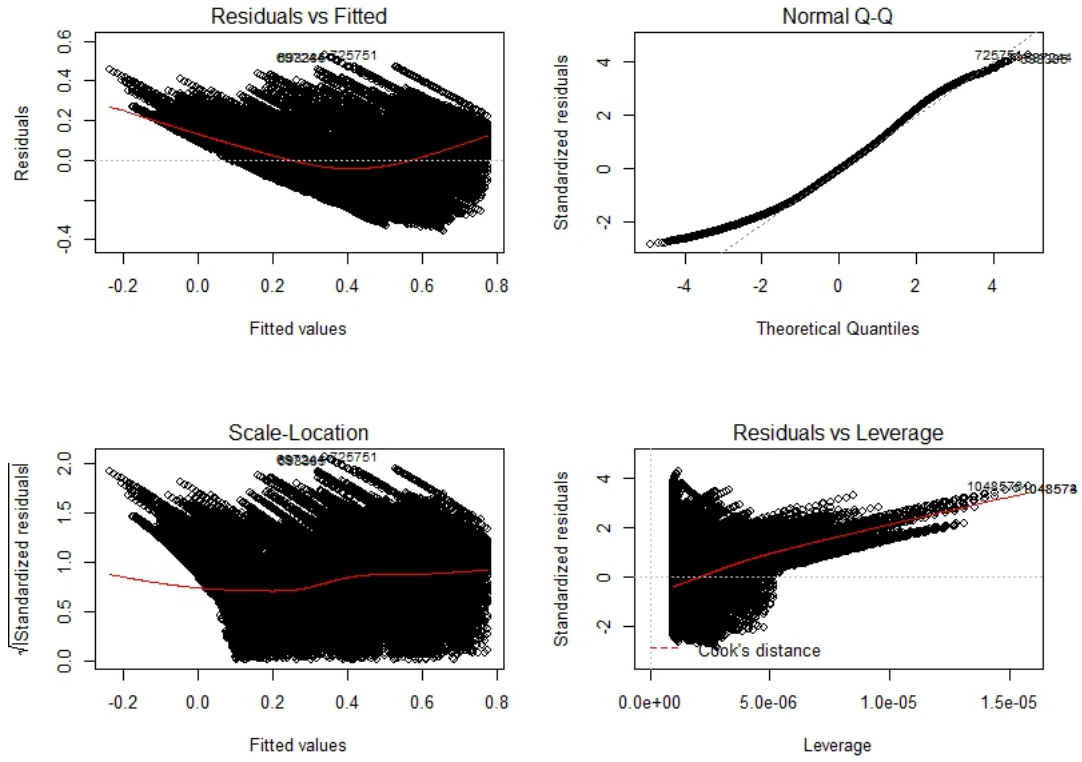


Figure 5.13: Reversed Residual plots for Test Area 1

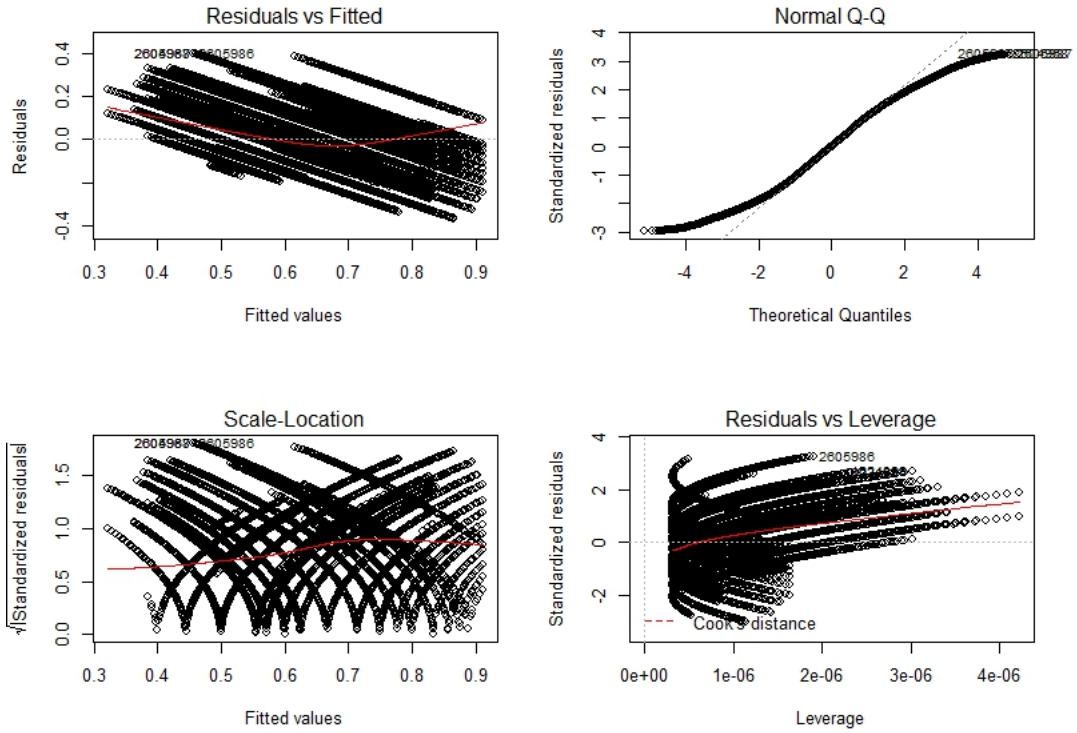


Figure 5.14: Reversed Residual Plots for Test Area 3

Findings: From the reversed residual plots, we can see that the errors/residuals are not normally distributed. The red lines in the Residuals vs Fitted plots show a curved shape, so that we conclude that the errors/residuals do not meet our linearity assumptions. This means that a similarity result is hard to predict given a distance. A malicious user will not be able to use pre-known distances to find the similarity pattern to locate other users by for example the triangulation methodology discussed in Qin, Patsakis, and Bouroche (2014).

5.2 Proximity Test with k-shingling and k-combination

Based on the privacy requirement stated in Section 2.2, the application server should only be allowed to compare the similarity of two data inputs but not be able to discover the original data sets. In addition, since cell tower identifiers are publicly known, in order to prevent a malicious user from sending fabricated data sets to deceive others, we modify the original data on the user side using k-shingling and k-combination, and evaluate the resulting distance versus similarity plots.

5.2.1 Experimental Results using k-shingling

To compare the outcome of using 2-shingling and 3-shingling with the original data, we use the same cell tower distributions in the test of similarity with distance using cell identifiers. The original test results, 2-shingling test results and 3-shingling test results are shown in the following for each test area:

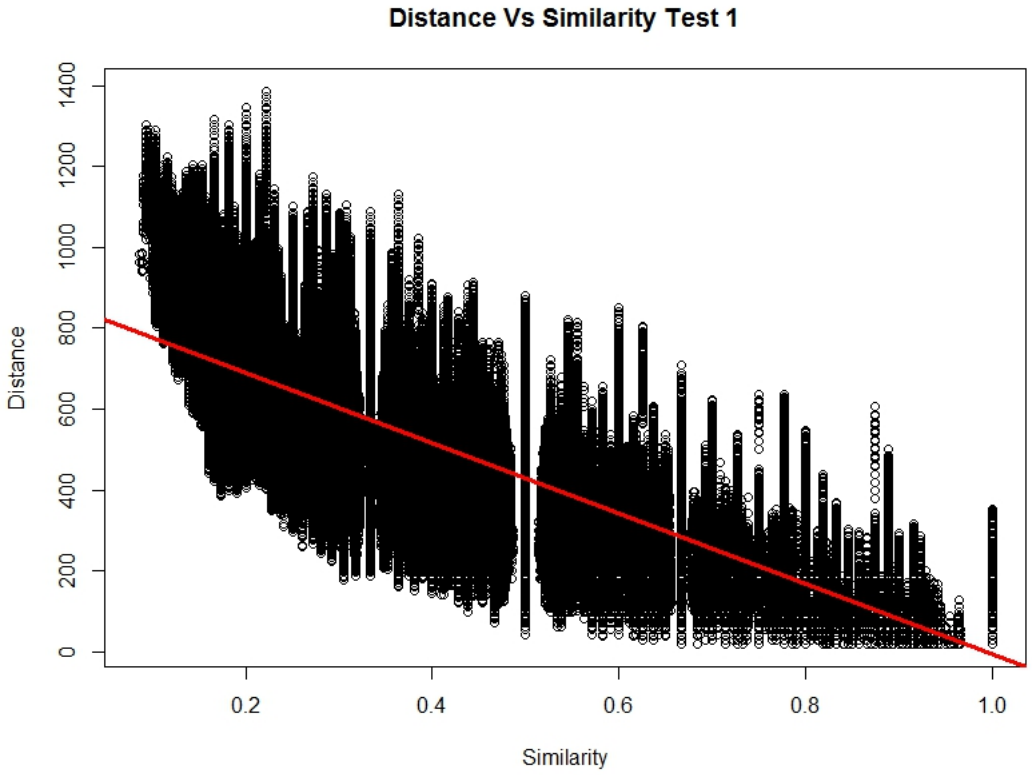


Figure 5.15: Distance vs. Similarity in Test Area 1

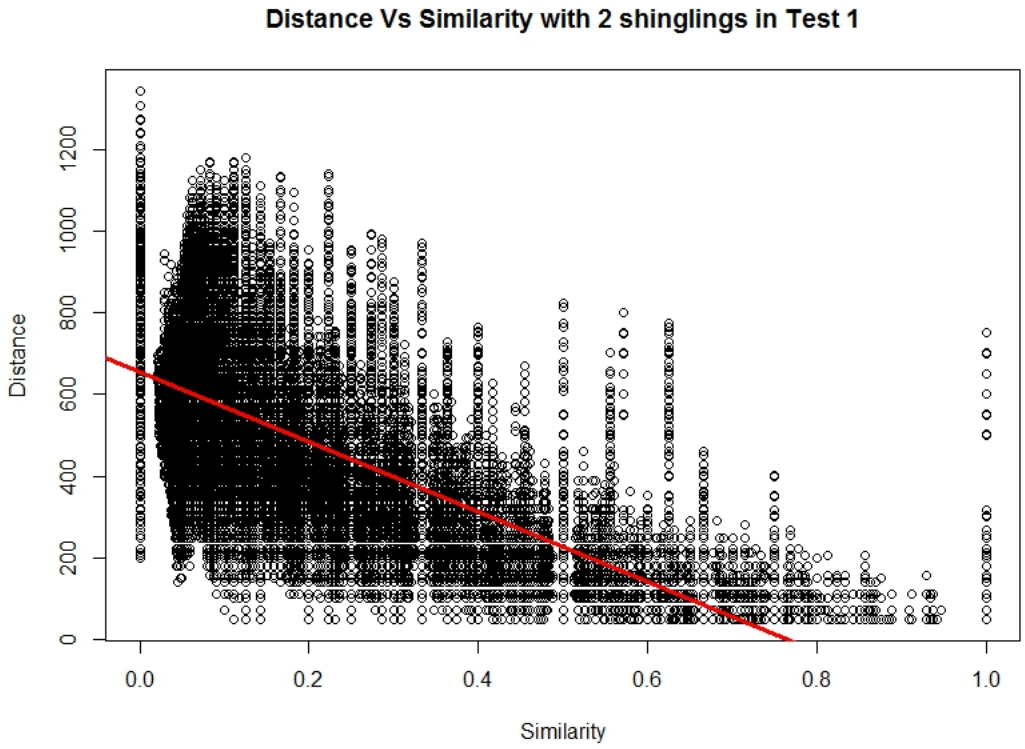


Figure 5.16: Distance vs. Similarity with 2-shingling in Test 1

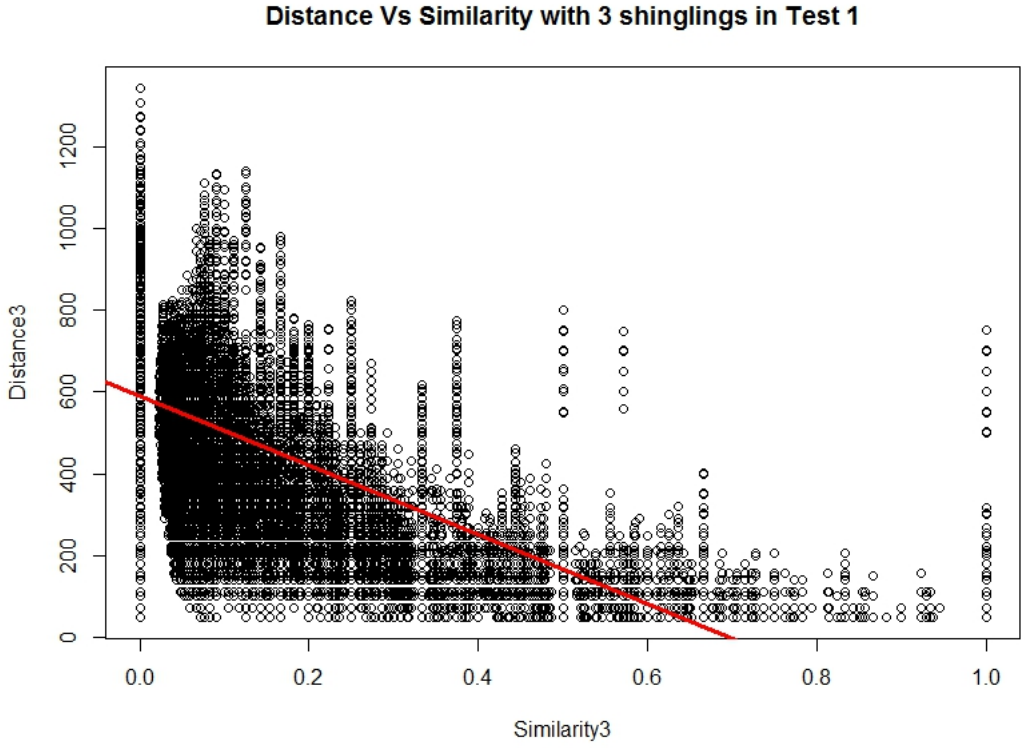


Figure 5.17: Distance vs. Similarity with 3-shingling in Test 1

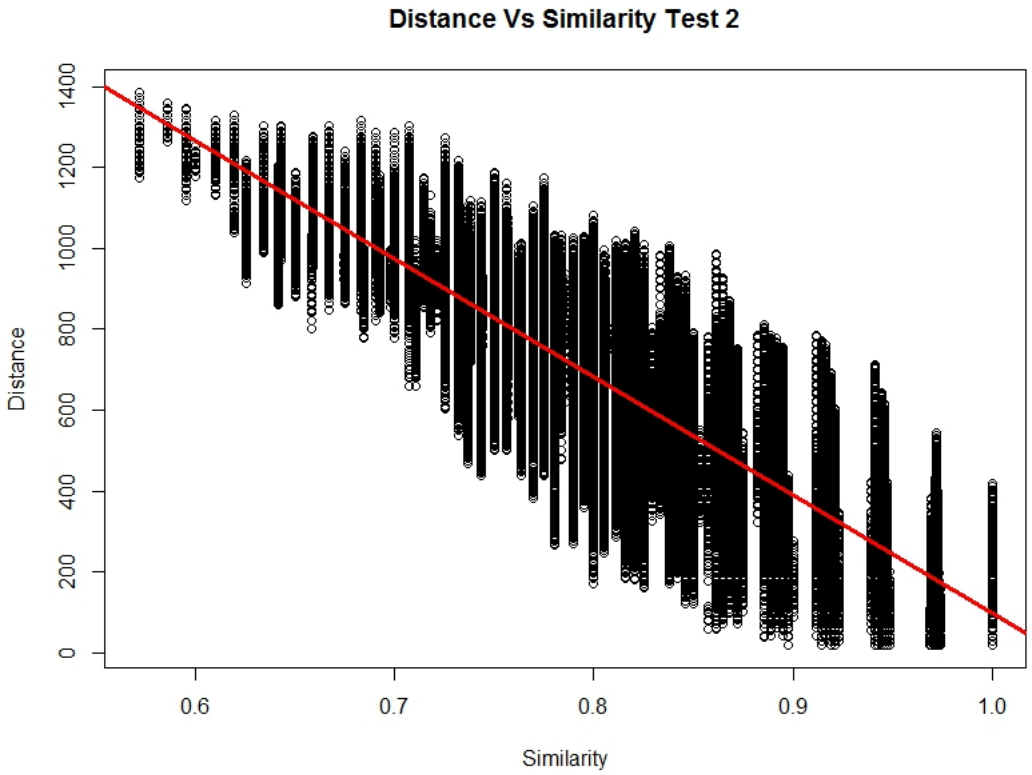


Figure 5.18: Distance vs. Similarity in Test Area 2

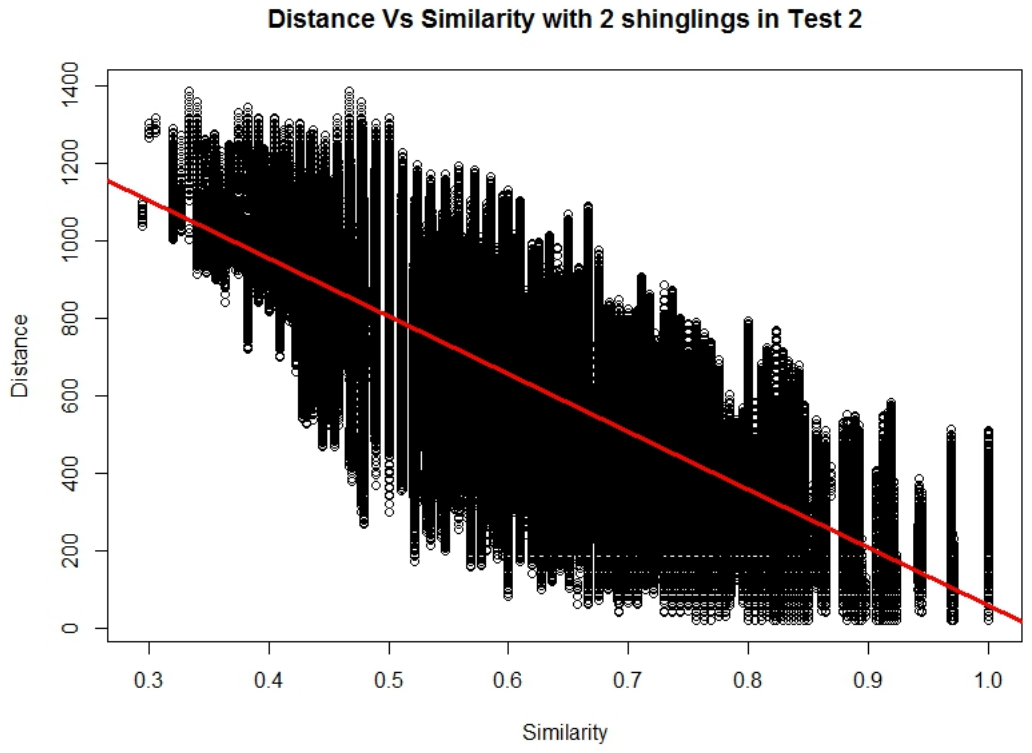


Figure 5.19: Distance vs. Similarity with 2-shingling in Test 2

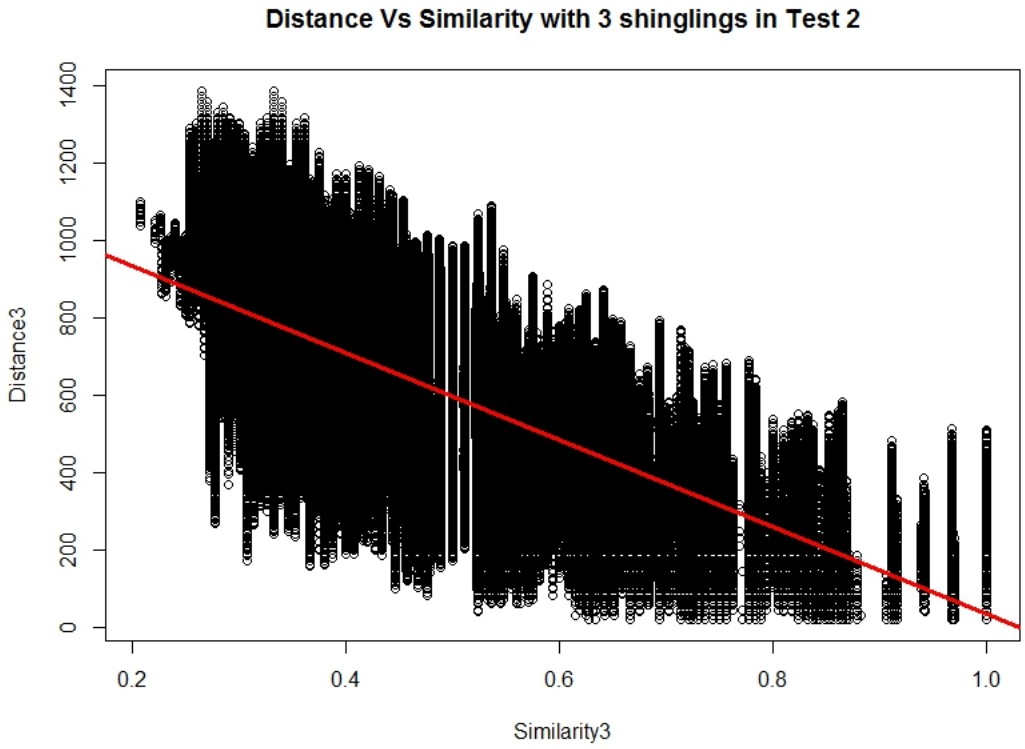


Figure 5.20: Distance vs. Similarity with 3-shingling in Test 2

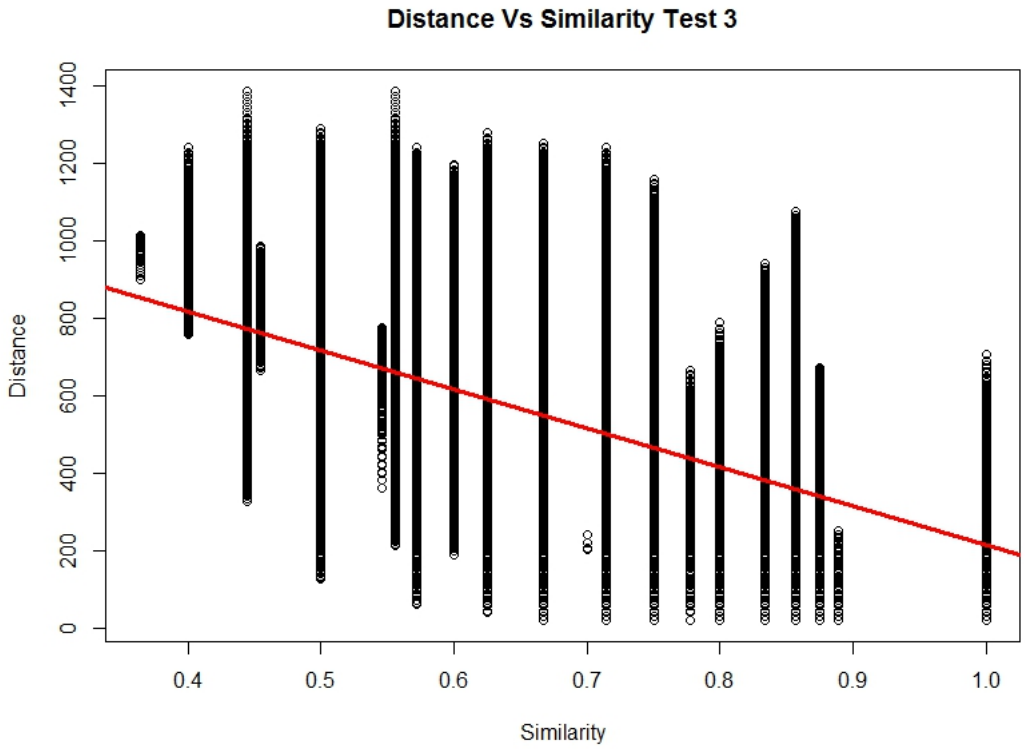


Figure 5.21: Distance vs. Similarity in Test Area 3

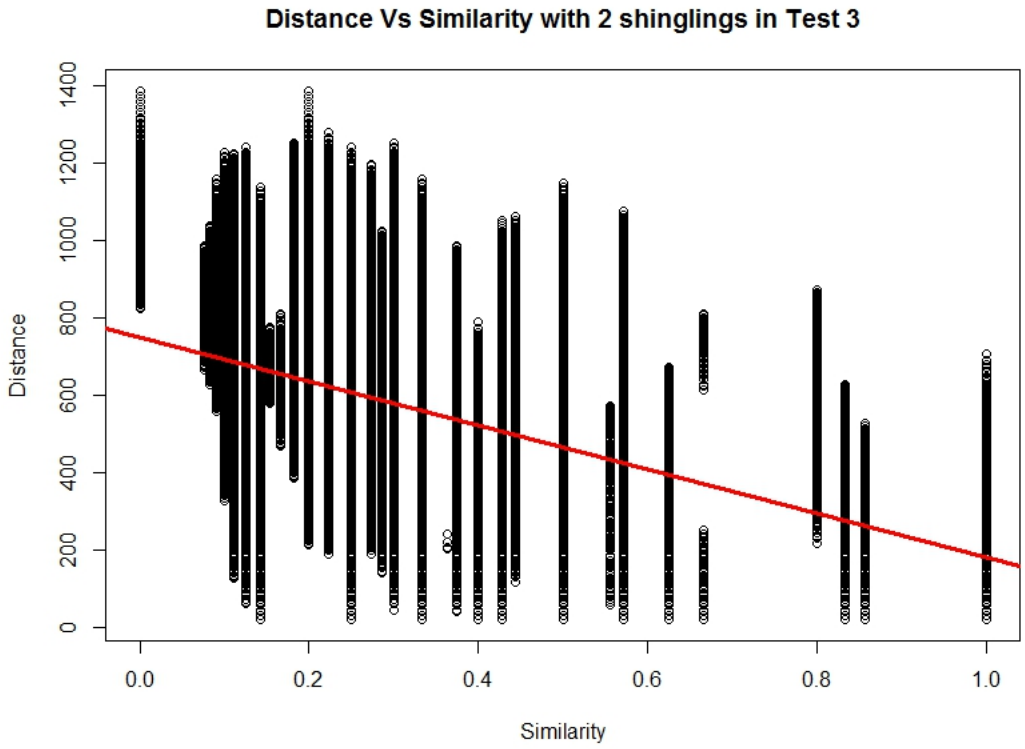


Figure 5.22: Distance vs. Similarity with 2-shingling in Test 3

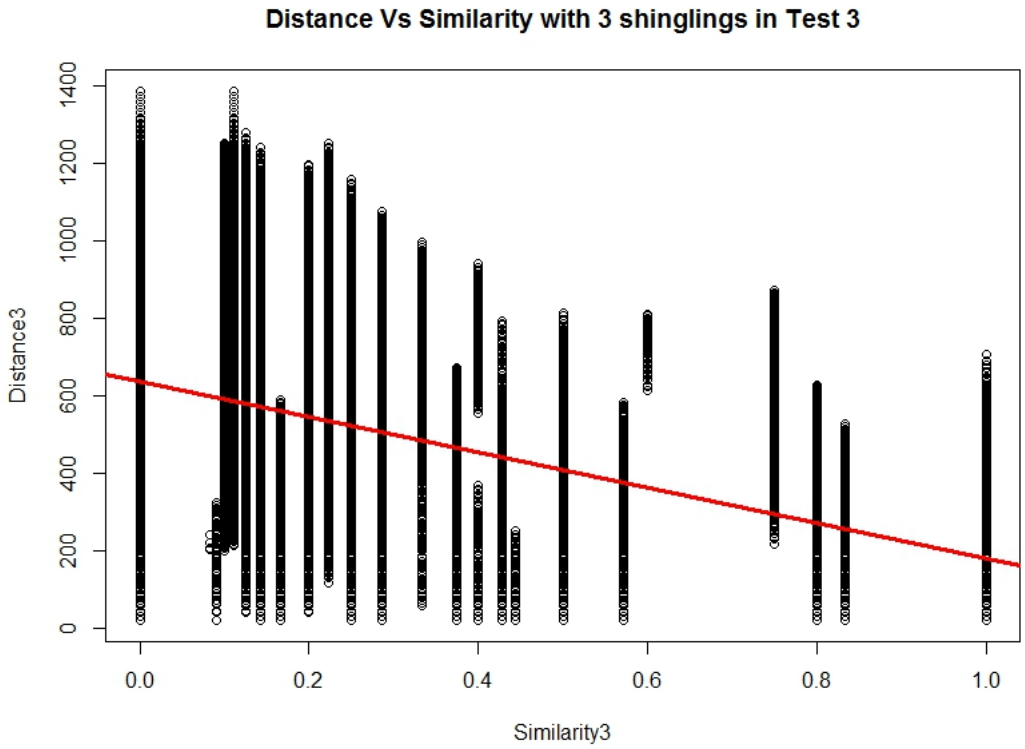


Figure 5.23: Distance vs. Similarity with 3-shingling in Test 3

Findings: From these figures, the whole graphs shift to the 0.0 coordinate when compared to the original plots of distance versus similarity in Test Area 1 and 3. This indicates that similarity results have been significantly reduced by using 2-shingling and 3-shingling. In addition, the variation in distance range increases from 2-shingling and 3-shingling. However, 3-shingling suffers a more significant affect than 2-shingling. The results show that it is difficult to indicate distance given a similarity after using 2-shingling and 3-shingling. Thus, this k-shingling methodology does not satisfy our requirements. This methodology only works well if the similar elements sit side by side, which must be consecutive. Hence, instead of implementing k-shingling, we subsequently modified the k-shingling principle to fit our scenario more appropriately. We call it K-combination.

5.2.2 Experimental Results using k-combination

The original test results, 2-combination test results and 3-combination test results are shown below for Test Area 1, 2 and 3:

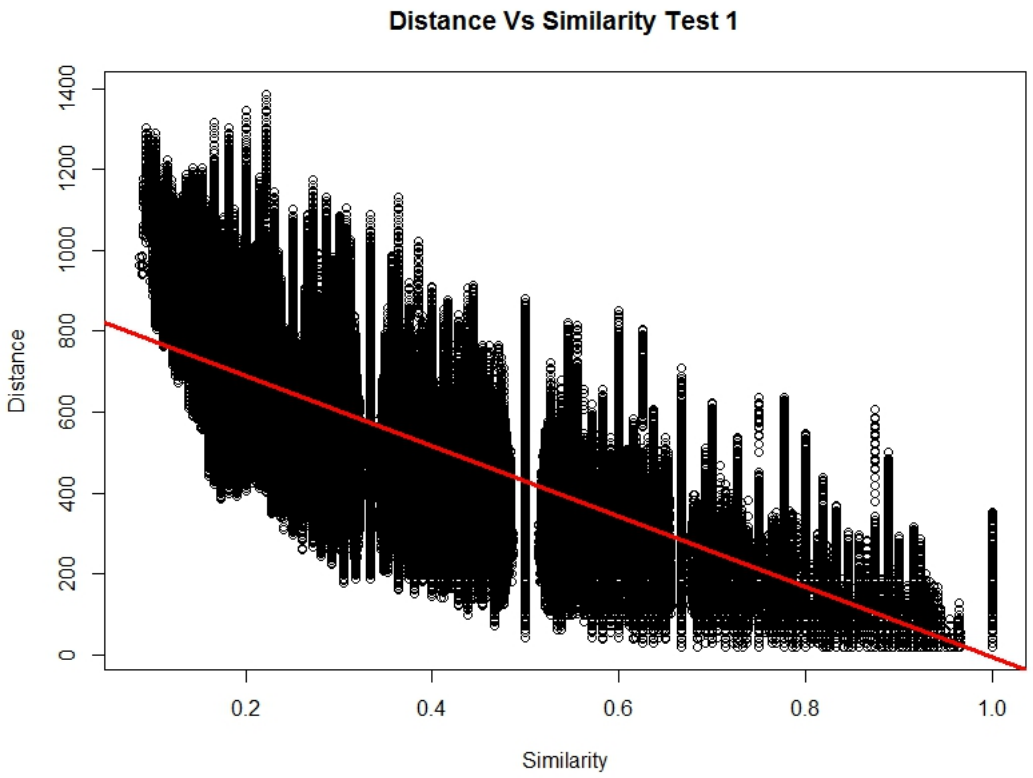


Figure 5.24: Distance vs. Similarity in Test Area 1

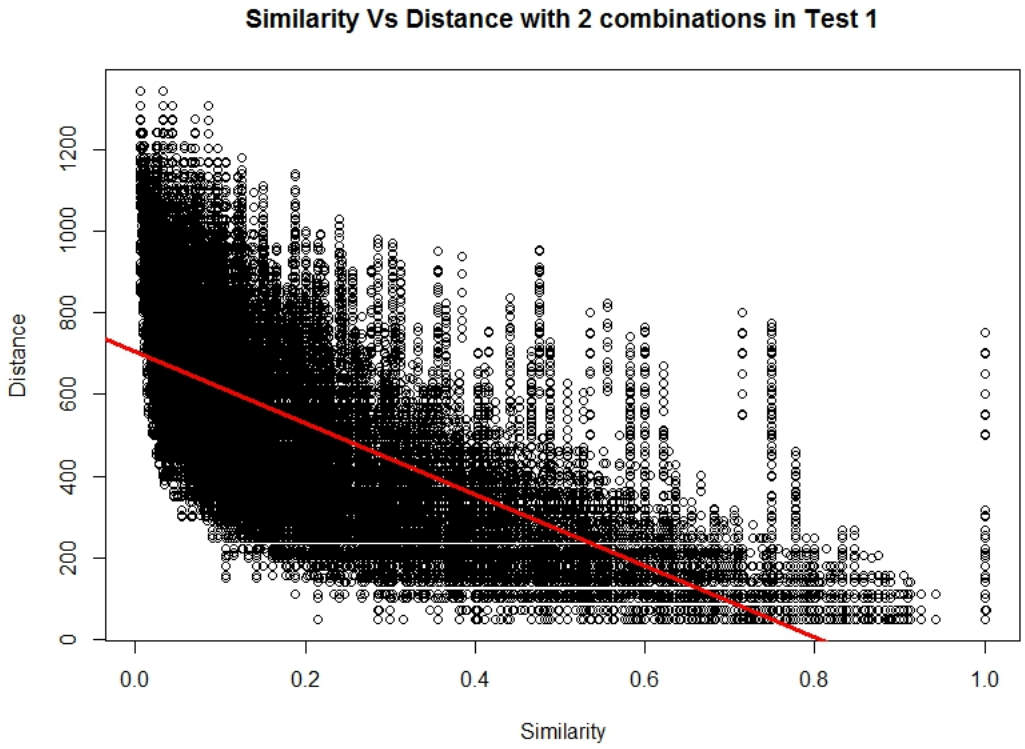


Figure 5.25: Distance vs. Similarity with 2-combination in Test 1

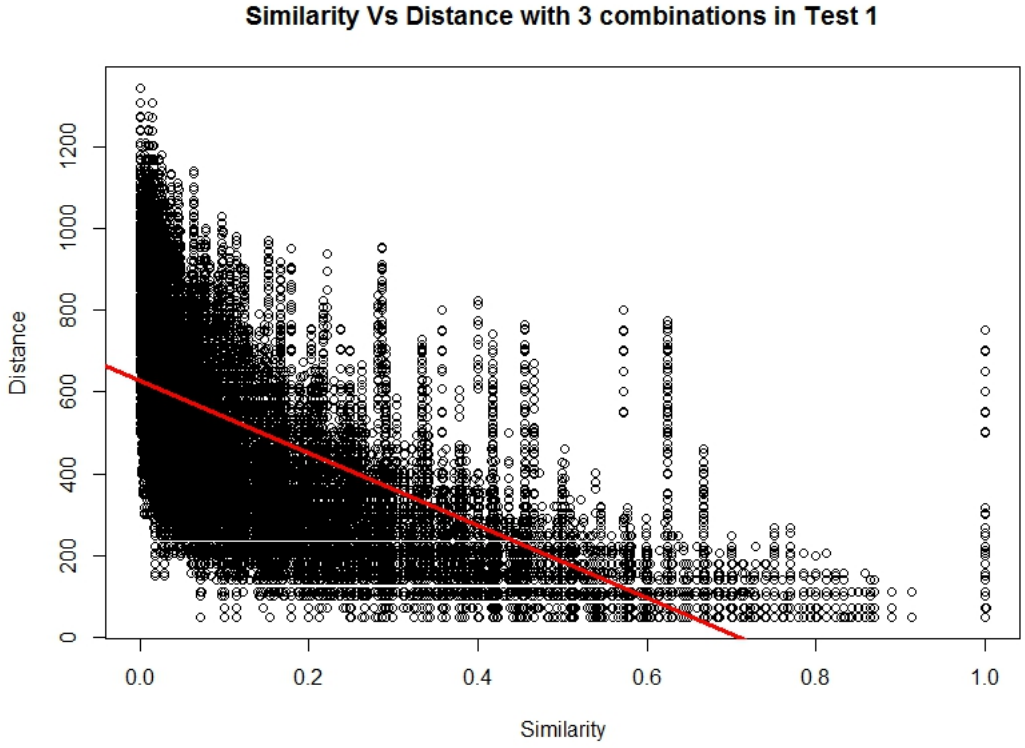


Figure 5.26: Distance vs. Similarity with 3-combination in Test 1

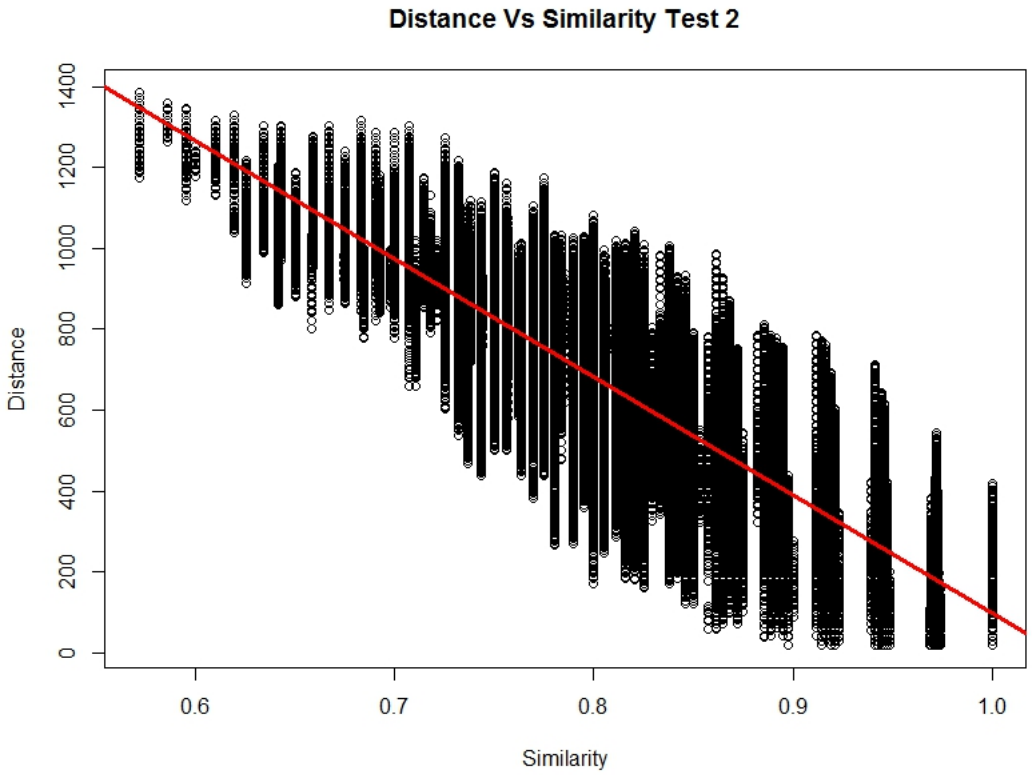


Figure 5.27: Distance vs. Similarity in Test Area 2

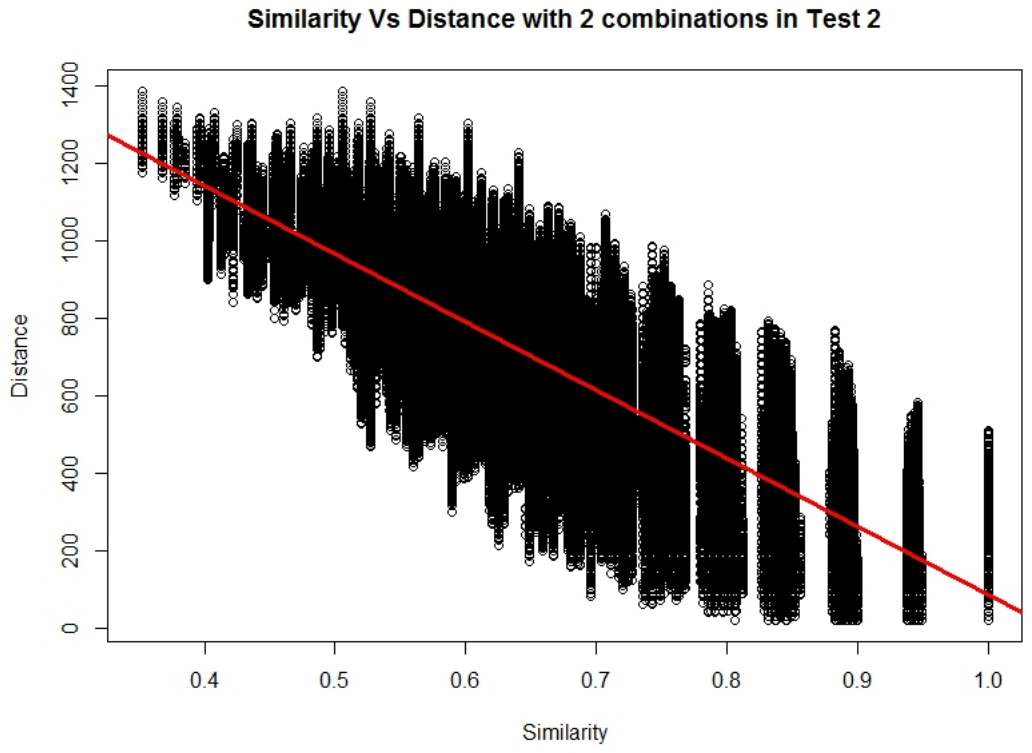


Figure 5.28: Distance vs. Similarity with 2-combination in Test 2

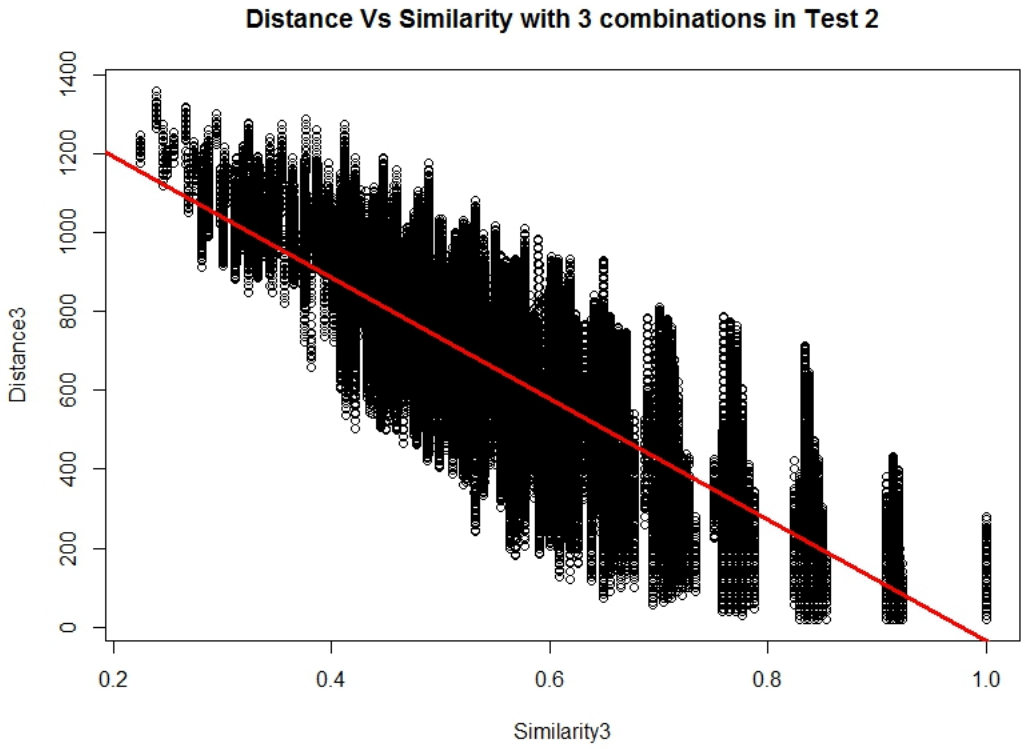


Figure 5.29: Distance vs. Similarity with 3-combination in Test 2

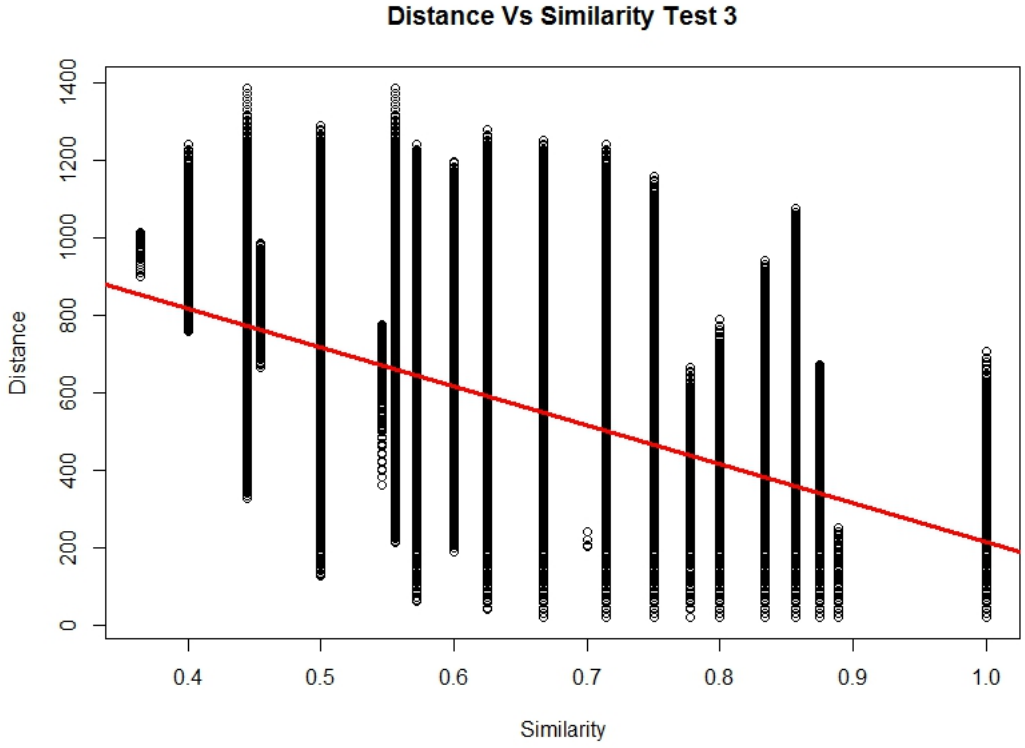


Figure 5.30: Distance vs. Similarity in Test Area 3

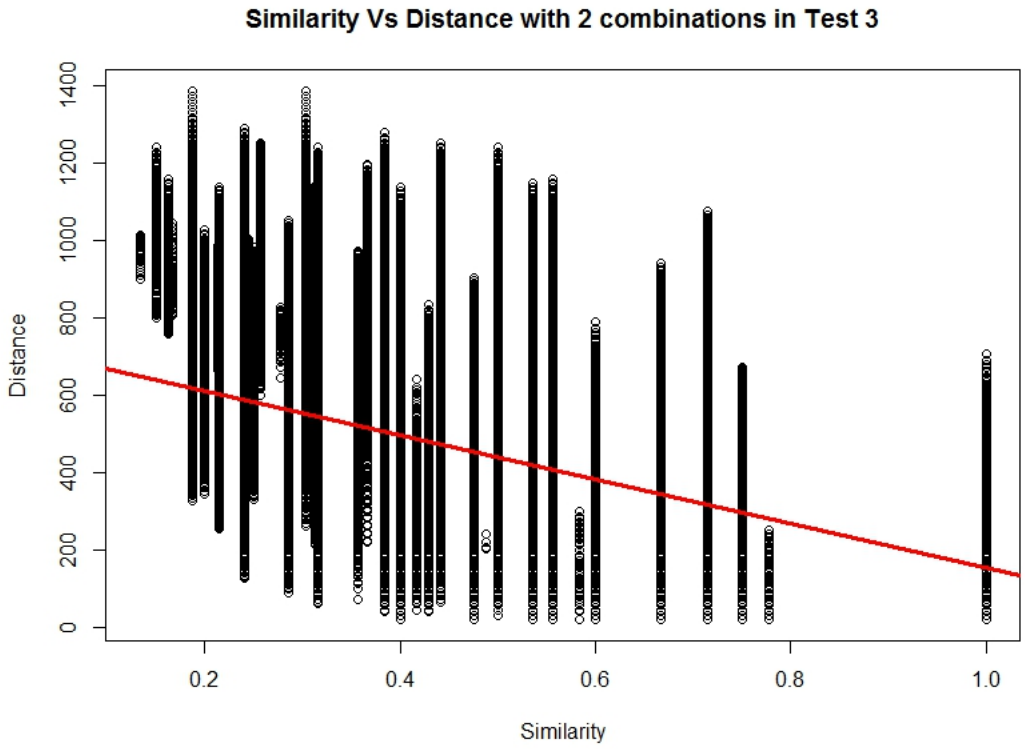


Figure 5.31: Distance vs. Similarity with 2-combination in Test 3

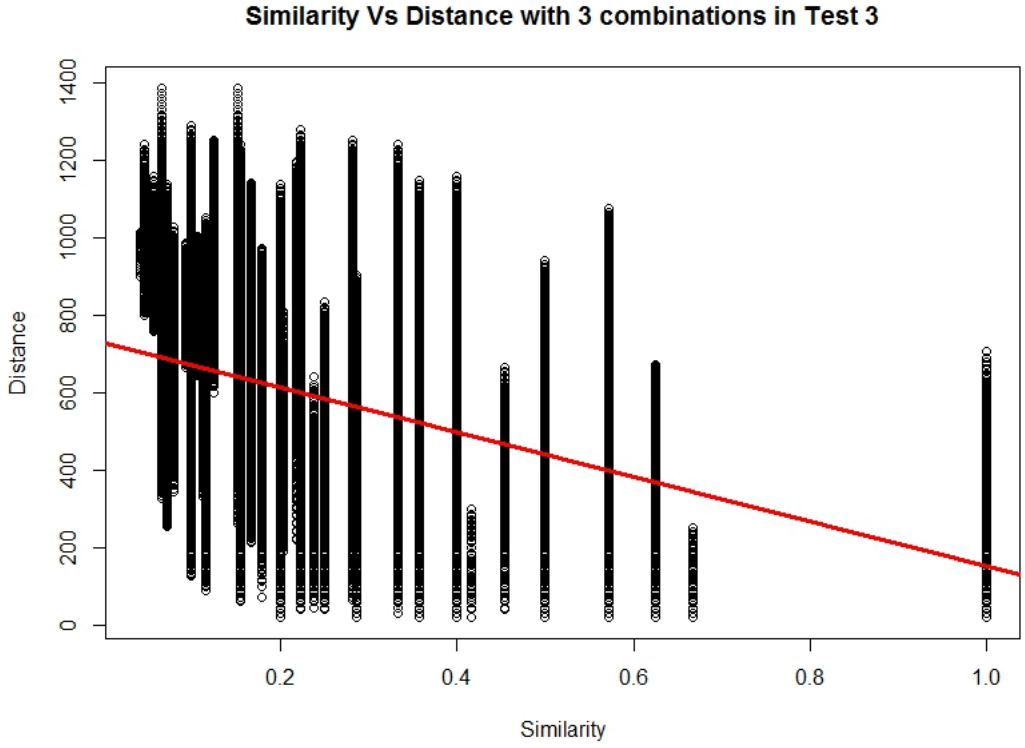


Figure 5.32: Distance vs. Similarity with 3-combination in Test 3

Findings: Taking the 200 metres distance line as the reference line, in Test Area 1, the distance between two points is larger than 200 metres when the similarity is smaller than 0.3 from the original test. After encryption with our 2-combination methodology, the similarity shifts from 0.3 to 0.2. This gives a more correct result compared to the result using the 2-shingling methodology. In Test Area 2, taking 200 metres as the reference line, the similarity result shifts from 0.8 to 0.7. In Test Area 3, the similarity result shifts from 0.45 to 0.25. The experiment results show that 2-combination methodology fits both our privacy requirements and application requirements.

5.3 Conclusion

In this section, we summarise our approach in three aspects, 1) resistance to privacy attacks, 2) satisfaction of location privacy requirements and 3) satisfaction of application requirements.

5.3.1 Evaluation of Resistance to Privacy Attacks

In our approach, the application server only receives a set of hashed values instead of exact coordinates or location regions. A malicious user will not be able to use the hashed values to discover other users' real locations directly. Thus, our approach is exempt from location homogeneity attacks, location distribution attacks and compromised trust third party attacks. However, due to the fact that cell tower identifiers are publicly known and that the hashed values are generated from cell tower identifiers, it is possible that the original set of cell tower identifiers might be obtained by brute force attacks. Then a compromised application server could possibly use those cell tower ID sets to find out in what region a target user might be. Nevertheless, to achieve this, a compromised application sever would need to download all the cell tower information and test what region is covered by a combination of cell tower identifiers. We assume that the total number of cell tower identifiers is n, thus, the combination of these cell tower identifiers is $n!$. The hypothesis test region is m square metres. Thus if we take 1 metre as the scale, the total test points will be m^2. If the server already knows each cell tower's coverage range and coordinate, to attack our algorithm and retrieve all the regions covered by each combination of cell towers identifiers, the calculation complexity is $O(n! * m^2)$. Besides, it is likely that more cell towers will be built in the city over time. It is very hard to follow the changes and recalculate all the possibilities. Even if a malicious user successfully reverse engineers the original set of cell tower identifiers, the target user's location is still covered

within 1000 metres from our experiment. Hence, our approach prevents context linking attacks, multiple query attacks, location tracking attacks, fake location attacks and so on. According to the different cell tower coverage ranges in different areas, using sets of cell towers as the location tag prevents against maximum movement boundary attacks as well. Thus, our approach is resilient from all the existing privacy attacks mentioned in Chapter 3.

5.3.2 Evaluation of Location Privacy Requirement

Based on our location privacy requirement from privacy of personal information, given information should not be directly or indirectly used to disclose users' social identity. However, in our literature review, location coordinates have been used to disclose certain users' social identities in different scenarios . Our methodology uses cell tower identifier sets instead of location coordinates, which protects users identities from being revealed to malicious users. Moreover, if malicious users know other users cell tower identifier sets, they cannot use such information to retrieve their real location, by using methods such as triangulation. Thus, our approach fits the first privacy requirement detailed above.

According to our location privacy requirement for privacy of personal communications, the users' locations should not be disclosed to application servers. In our methodology, the original data set obtained from each mobile user is hashed using the k-combination method and sent to application servers encrypted. It is difficult for a server to decode or reverse engineer the original data sets in a short period of time. Assuming that an application server knows the original data sets, from our experiments, two points that have the exact same data set could be in a distance range of 100 metres to 1000 metres. Hence, we conclude that the user's location privacy is protected to the level of 1000 metres even if an application server knows the original data set. Thus, our approach fits the second location privacy requirement.

For the location privacy requirements of privacy of person and privacy of personal behaviour, the users' location information should only be controlled by users themselves and their physical body should be protected from harm by malicious users. In our methodology, the collected cell tower identifier sets will only be stored for the period of time when the user wants to share their information. In addition, based on our test results which show that the highest similarity of two data sets indicates a distance between two users in a range of 100 metres to 1000 metres, users would be able to avoid physical harm from malicious users. This shows that our approach fits the last two location privacy requirements

as well.

5.3.3 Evaluation of Application Requirement

From our experimental results, we can see that the definition of 'close' is varied by the density and radio range of cell towers. Due to the density of buildings, if two users are both in the city centre, we should consider that they are close when the distance between the two of them is less than 500 metres. If two users are far out of city center, since the area has less building to cover the users, we should consider they are close when the distance between two of them are less than 1000 metres or more. In our approach, the geographic proximity results have been dynamically scaled by following the density of mobile users. This will solve the false positive and negative results problem arising with predefined cells. When two of the users are close to each other, the system will show a high similarity rate. On the contrary, our system is able to define the proximity by similarity.

Chapter 6

Conclusions and Future Work

The main focus of this thesis was the presentation of a cell tower identifier dataset based location privacy preserving approach called k-combination. This method provides a self-organizing location obfuscation solution to ensure users' location information is protected in LBSN applications. This concluding chapter summarises the most significant achievements of the work described herein and assess its contribution to the body of knowledge within the field of computer science. In addition, some suggestions for future work are outlined.

6.1 Achievements

The motivation for the work described in this thesis arose from the increasing location privacy concerns of dating applications used in our daily lives. As dating applications are becoming more popular, location privacy threats to end users are increasing at the same time. By running a practical fake location attack with trilateration, we found serious location privacy leaks from some dating applications. Instead of developing a location privacy solution for dating applications alone, we built a model which could work with most LBSN applications.

A review of existing work on state-of-the-art solutions for participatory sensing applications and location-based applications, has shown that none of the existing work is suitable for addressing the challenges of protecting location privacy against both malicious users and application servers, which also can be considered as administrators. Those existing solutions all suffered from different privacy attacks to a certain degree. It was a big challenge to find a solution that could resist all existing privacy attacks and was also deployable to mobile devices. Especially, one key requirement is that the application server

must be able to provide a proximity function but without knowing each user's location.

Our k-combination approach allows a server to check if two users are close without taking the users' actual location coordinates. Moreover, this work provides a self-organizing location obfuscation solution and an extra layer to prevent malicious users from retrieving the original dataset. K-combination can be easily employed to mobile devices and it can satisfy LBSN application requirement to a large degree. To the best of our knowledge, we are the first researchers to use cell tower identifier datasets as a type of location tag.

6.2 Future Work

As is always the case in research, and particularly with this work that investigated a new research direction, many issues are worthy of more detailed investigation.

K-combination presents a number of results that are explained, but not formally proven. Our work is based on data obtained from the OpenCellID project. A significant extension to this work would be to test it in a real mobile network situation. The patterns for estimating distance in relation to similarity could be different when analysed in a real world scenario. In addition, our system model has not been implemented in mobile devices for testing LBSN applications. It would be fruitful and beneficial to extend the research further by completing such an implementation and analysing the performance.

Finally, our experiment is limited by the number of samples we have used. More future work could be conducted to test larger datasets in different countries. This may reveal patterns that could be taking into account in the k-combination calculation to increase the accuracy of similarity calculation. This work concentrated on solving the problem of location privacy in LBSN application but it does not provide a solution to users faking their locations. However, a location verification layer could be added into our protocol using long range wireless broadcasting to make the check-in function more reliable. It would be a worthwhile endeavour to test this added feature in any future analyses.

6.3 Conclusion

This chapter summarised the motivations for this research, as well as the most significant achievements of the work described in this thesis. In particular, it outlined how this work contributes knowledge towards the state of the art in solutions to issues relating to location privacy. In addition, several suggestions for future work were presented, such as implementation on mobile devices.

Bibliography

Abul, Osman, Francesco Bonchi, and Mirco Nanni (2008). "Never walk alone: Uncertainty for anonymity in moving objects databases". In: *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. Ieee, pp. 376–385.

Ardagna, Claudio et al. (2007). "Location privacy protection through obfuscation-based techniques". In: *Data and Applications Security XXI*, pp. 47–60.

Bartoli, A. et al. (2012). "On the Ineffectiveness of Todays Privacy Regulations for Secure Smart City Networks". In: *third IEEE International Conference on Smart Grid Communications*.

Beresford, Alastair R and Frank Stajano (2004). "Mix zones: User privacy in location-aware services". In: *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*. IEEE, pp. 127–131.

Chang, Wei, Jie Wu, and Chiu C Tan (2011). "Enhancing mobile social network privacy". In: *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. IEEE, pp. 1–5.

Chow, Chi yin and Mohamed F. Mokbel (2007). *Enabling Private Continuous Queries For Revealed User Locations*.

Christin, Delphine et al. (2011). "A survey on privacy in mobile participatory sensing applications". In: *Journal of systems and software* 84.11, pp. 1928–1946.

Dingledine, R., N. Mathewson, and P. Syverson (2004). "Tor: the second-generation onion router". In: *the 13th Conference on USENIX Security Symposium(USENIX Security)*, pp. 21–38.

Dürr, Frank, Pavel Skvortsov, and Kurt Rothermel (2011). "Position sharing for location privacy in non-trusted systems". In: *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*. IEEE, pp. 189–196.

Gambs, Sébastien, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez (2014). "De-anonymization attack on geolocated data". In: *Journal of Computer and System Sciences* 80.8, pp. 1597–1614.

Gedik, Bugra and Ling Liu (2005). "Location privacy in mobile systems: A personalized anonymization model". In: *Distributed computing systems, 2005. ICDCS 2005. Proceedings. 25th IEEE international conference on*. IEEE, pp. 620–629.

Ghinita, Gabriel et al. (2008). "Private queries in location based services: anonymizers are not necessary". In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, pp. 121–132.

Ghinita, Gabriel et al. (2009). "Preventing velocity-based linkage attacks in location-aware applications". In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, pp. 246–255.

Gkoulalas-divanis, Aris, Vassilios S. Verykios, and Mohamed F. Mokbel (2009). *Identifying Unsafe Routes for Network-Based Trajectory Privacy*.

Golle, Philippe and Kurt Partridge (2009). "On the anonymity of home/work location pairs". In: *Pervasive Computing*. Springer, pp. 390–397.

Greschbach, Benjamin and Sonja Buchegger (2012). "Friendly surveillance–a new adversary model for privacy in decentralized online social networks". In: *Proceedings of the 5th Interdisciplinary Conference on Current Issues in IT Security*, pp. 5–206.

Gruteser, Marco and Dirk Grunwald (2003). "Anonymous usage of location-based services through spatial and temporal cloaking". In: *Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM, pp. 31–42.

Gruteser, Marco, Dirk Grunwalddepartment, and Computer Science (2003). "Anonymous usage of location-based services through spatial and temporal cloaking". In: *ACM Int'l Conf.Mobile Systems,Applications,and Services*, pp. 31–42.

Hallgren, Per, Martin Ochoa, and Andrei Sabelfeld (2015). "InnerCircle: A parallelizable decentralized privacy-preserving location proximity protocol". In: *Privacy, Security and Trust (PST), 2015 13th Annual Conference on*. IEEE, pp. 1–6.

He, Wenbo, Xue Liu, and Mai Ren (2011). "Location cheating: A security challenge to location-based social network services". In: *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*. IEEE, pp. 740–749.

Humphreys, Lee (2007). "Mobile social networks and social practice: A case study of Dodgeball". In: *Journal of Computer-Mediated Communication* 13.1, pp. 341–360.

Khuong Vu, Rong Zheng and Jie Gao (2012). "Efficient Algorithms for K-Anonymous Location Privacy in Participatory Sensing". In: *INFOCOM,2012 Proceedings IEEE*, pp. 2399–2407.

Kido, Hidetoshi, Yutaka Yanagisawa, and Tetsuji Satoh (2005). "An anonymous communication technique using dummies for location-based services". In: *Pervasive Services, 2005. ICPS'05. Proceedings. International Conference on*. IEEE, pp. 88–97.

Krumm, John (2007). "Inference attacks on location tracks". In: *Pervasive computing*, pp. 127–143.

Li, Nan and Guanling Chen (2010). "Sharing location in online social networks". In: *IEEE network* 24.5.

Li, Po-Yi et al. (2008). "A cloaking algorithm based on spatial networks for location privacy". In: *Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC'08. IEEE International Conference on*. IEEE, pp. 90–97.

Lin, Zi and Denis Foo Kune (2012). "Efficient Private Proximity Testing with GSM Location Sketches". In: *32nd International Cryptology Conference*.

Liu, Kun, Hillol Kargupta, and Jessica Ryan (2006). "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining". In: *IEEE Transactions on knowledge and Data Engineering* 18.1, pp. 92–106.

Liu, Ling (2007). "From data privacy to location privacy: models and algorithms". In: *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, pp. 1429–1430.

Machanavajjhala, Ashwin et al. (2007). "l-diversity: Privacy beyond k-anonymity". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1, p. 3.

Madden, Mary et al. (2013). "Teens and Mobile Apps Privacy". In: *Pew Internet and American Life Project*.

Mascetti, Sergio et al. (2009). "Privacy-Aware Proximity Based Services." In: *Mobile Data Management*. IEEE Computer Society, pp. 31–40.

Mokbel, Mohamed F (2007). "Privacy in Location-based Services: State-of-the-art and Research Directions". In: *Mobile Data Management, 2007 International Conference on*. IEEE, pp. 228–228.

Narayanan, Arvind et al. (2011). "Location Privacy via Private Proximity Testing." In: *NDSS*. Vol. 11.

Noulas, Anastasios et al. (2011). "An empirical study of geographic user activity patterns in foursquare." In: *ICwSM* 11, pp. 70–573.

Pan, Xiao, Jianliang Xu, and Xiaofeng Meng (2012). "Protecting Location Privacy against Location-Dependent Attack in Mobile Services". In: *Knowledge and Data Engineering* 10, pp. 1506–1519.

Piro, Chris, Clay Shields, and Brian Neil Levine (2006). "Detecting the sybil attack in mobile ad hoc networks". In: *Securecomm and Workshops, 2006*. IEEE, pp. 1–11.

Polakis, Iasonas et al. (2015). "Where's Wally?: Precise user discovery attacks in location proximity services". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, pp. 817–828.

Qin, Guojun, Constantinos Patsakis, and Mélanie Bouroche (2014). "Playing hide and seek with mobile dating applications". In: *IFIP International Information Security Conference*. Springer, pp. 185–196.

Qiu, Di et al. (2009). "Robust location tag generation from noisy location data for security applications". In: *The Institute of navigation international technical meeting*, pp. 586–597.

Saldamli, Gokay et al. (2013). "Private Proximity Testing with an Untrusted Server". In: *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*. WiSec '13. Budapest, Hungary: ACM, pp. 113–118. ISBN: 978-1-4503-1998-0.

Schnorr, Claus Peter and Markus Jakobsson (2000). "Security of signed ElGamal encryption". In: *Advances in CryptologyASIACRYPT 2000*. Springer, pp. 73–89.

Shi, Jing et al. (2010). "Prisense: privacy-preserving data aggregation in people-centric urban sensing systems". In: *INFOCOM, 2010 Proceedings IEEE*. IEEE, pp. 1–9.

Shilton, Katie (2009). "Four billion little brothers?: Privacy, mobile phones, and ubiquitous data collection". In: *Communications of the ACM* 52.11, pp. 48–53.

Shokri, Reza et al. (2011). "Quantifying location privacy". In: *Security and privacy (sp), 2011 ieee symposium on*. IEEE, pp. 247–262.

Šikšnys, Laurynas et al. (2009). "A location privacy aware friend locator". In: *Advances in Spatial and Temporal Databases*. Springer, pp. 405–410.

Siksnys, Laurynas et al. (2010). "Private and Flexible Proximity Detection in Mobile Social Networks." In: *Mobile Data Management*. IEEE Computer Society, pp. 75–84.

Solanas, Agusti, Francesc Sebé, and Josep Domingo-Ferrer (2008). "Micro-aggregation-based heuristics for p-sensitive k-anonymity: one step beyond". In: *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*. ACM, pp. 61–69.

Symeonidis, Panagiotis, Dimitrios Ntempos, and Yannis Manolopoulos (2014). "Location-Based Social Networks". In: *Recommender Systems for Location-based Social Networks*. Springer, pp. 35–48.

Talukder, Nilothpal and Sheikh Iqbal Ahamed (2010). "Preventing multi-query attack in location-based services". In: *Proceedings of the third ACM conference on Wireless network security*. ACM, pp. 25–36.

Terrovitis, Manolis and Nikos Mamoulis (2008). "Privacy preservation in the publication of trajectories". In: *Mobile Data Management, 2008. MDM'08. 9th International Conference on*. IEEE, pp. 65–72.

Ulm, Michael, Peter Widhalm, and Norbert Brändle (2015). "Characterization of mobile phone localization errors with OpenCellID data". In: *Advanced Logistics and Transport (ICALT), 2015 4th International Conference on*. IEEE, pp. 100–104.

Wernke, Marius et al. (2014). "A Classification of Location Privacy Attacks and Approaches". In: *Personal Ubiquitous Comput.* 18.1, pp. 163–175. ISSN: 1617-4909. DOI: 10.1007/s00779-012-0633-z. URL: http://dx.doi.org/10.1007/s00779-012-0633-z.

Xu, Jianhua et al. (2015). "Applications of mobile social media: WeChat among academic libraries in China". In: *The Journal of Academic Librarianship* 41.1, pp. 21–30.

Zhang, Chengyang and Yan Huang (2009). "Cloaking locations for anonymous location based services: a hybrid approach". In: *GeoInformatica* 13.2, pp. 159–182.

Zheng, Yao et al. (2012). "SHARP: Private Proximity Test and Secure Handshake with Cheat-Proof Location Tags." In: *ESORICS*. Vol. 7459. Lecture Notes in Computer Science. Springer, pp. 361–378.

Zickuhr, Kathryn (2013). "Location-Based Services". In: *Pew Internet and American Life Project*.