Location Assurance and Privacy in Location-based Services

Xihui CHEN

Supervisor:

Prof. Dr. Sjouke Mauw (University of Luxembourg)

Daily advisor:

Dr. Jun Pang (University of Luxembourg)

The author was employed at the University of Luxembourg and supported by the Fonds National de la Recherche Luxembourg (FNR) in the project "Secure and Private Location Proofs: Architecture and Design for Location Based Services" (reference SECLOC 794361). Part of the research in this thesis is funded by the European Space Agency (ESA) project "Developing a Prototype of Localisation Assurance Service Provider (LASP)" (contract number 000102584-10-NL-HE), in collaboration with itrust consulting s. à r. l., Luxembourg and the Interdisciplinary Centre for Security, Reliability and Trust (SnT) of the University of Luxembourg.

UNIVERSITÉ DU LUXEMBOURG

PhD-FSTC-2014-16
The Faculty of Sciences, Technology and Communication

DISSERTATION

Presented on 20/06/2014 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG
EN INFORMATIQUE

by

Xihui CHEN

Born on 29 April 1982 in Shandong (China)

Location Assurance and Privacy in Location-based Services

Dissertation defense committee

Dr. Yves Le Traon, chairman
Professor, Université du Luxembourg

Dr. Gabriele Lenzini
Université du Luxembourg

Dr. Ling Liu
Professor, Georgia Institute of Technology

Dr. Sjouke Mauw, dissertation supervisor
Professor, Université du Luxembourg

Dr. Catuscia Palamidessi
Professor, INRIA

Dr. Jun Pang, vice-chairman
Université du Luxembourg

Summary

Nowadays, most mobile devices are equipped with positioning capabilities thanks to the free and ubiquitous access to global navigation satellite systems (GNSS) such as the global positioning system (GPS). With such mobile devices, people are able to obtain their precise locations, which in turn leads to a wide range of location-based services (LBS). Through an LBS, a user can acquire information customised to his locations. However, the vulnerabilities of GNSS systems and the exposure of information such as locations and queries in LBS requests impose a strong need from users on security. In this thesis, we study two security requirements in LBSs: *location assurance* and *privacy*.

Location assurance expresses users' requirement for trustworthy locations in terms of correctness and precision. To make use of LBSs, users rely on positioning systems to calculate their locations. However, these systems are vulnerable to attacks which can maliciously affect the calculation of locations. For instance, GPS suffers from *spoofing* attacks which can fool GPS receivers to calculate incorrect locations by sending false satellite signals. These attacks degrade the precision of calculated locations or even result in incorrect ones.

Privacy addresses users' concern about personal information leakage in LBSs. To make use of an LBS, a user needs to expose his location and query to an LBS provider. When such information is abused, for instance, by attackers who eavesdrop the public channel used for sending LBS requests or malicious LBS providers, users' private information such as habits and occupations will be inferred. Two types of privacy in LBSs have been identified in the literature and will be discussed in this thesis: *query privacy* and *location privacy*, which are related to privacy leakage from what users asked and where users went, respectively.

The first part of this thesis considers the vulnerability of GNSS systems to spoofing attacks and presents a trust framework to detect spoofing by evaluating the integrity of GNSS signals. The framework combines existing spoofing detection methods to generate an overall quantitative evaluation of the integrity of received signals. Based on this evaluation, users can determine the extent to which they can trust their locations. We implement a prototype based on our framework and develop a public service called *location assurance certification*. In this service, a trusted agent is introduced to certify uses' locations according to the integrity of their received signals.

The second part of this thesis discusses the protection of query privacy when the adversary has access to an increasing amount of contextual information. First, we propose a probabilistic framework. In this framework, we formally define the attacks which enable the adversary to infer the issuers of LBS queries by exploring

various contextual information. Second, we make use of two types of contextual information, *users profiles* and *query dependency*, to instantiate our framework. User profiles have not been deeply studied in the literature while dependency between queries has not been considered yet. Third, we propose a series of query privacy metrics. These metrics not only measure query privacy from different perspectives but also enable users to express their requirements for query privacy flexibly and precisely. In order to meet users' requirement on query privacy, we develop new protection mechanisms against attacks exploring contextual information.

The third part of this thesis addresses location privacy. So as to protect users' location privacy, many location privacy preserving methods (LPPM) have been proposed. A user will make use of them to break the link between his identity and his locations when requesting LBSs. We propose a new attack on location privacy based on the adversary's observation on users' locations protected by LPPMs. Compared to existing attacks which target at where users went, our attack provides the adversary with sufficient information to infer what users did, i.e., their activities. Specifically, through our attack, the adversary learns the places where users performed activities and their beginning and ending time of each activity. To achieve this goal, we explore the patterns of users with respect to movements and requesting LBSs, i.e., user profiles. So as to capture users' behaviour naturally and ensure the accuracy of the temporal information, i.e., beginning time and ending time of activities, we propose a new model for user profiles which models time as continuous.

Acknowledgments

Doing PhD is like taking a trip to the unknown. It starts with excitement, proceeds with fear of failure, and finishes with great joy. I am lucky to have been surrounded by lots of nice people who have been helping me overcome the fear and accomplish the trip ever since they met me. I would like to thank all of them for their selfless and sincere support.

I owe my deepest gratitude to my daily advisor Jun Pang. Our collaboration started six years ago. Back then, I was still an exchanging master student from Shandong University. His passion for research made research a cool thing, which attracts me to follow his steps. He taught me how to do research, how to write papers and most importantly, how to be a better person. However, to be as organised as him, I still have a long way to go. His strict supervision is tough but makes me grow faster and obtain more and better results than I should have deserved. I would remember the prosperous after-lunch coffee discussions and all great food we tried together. I also want to say sorry to him for putting him in the pressure of catching deadlines from time to time.

My thesis will have never been possible without Sjouke Mauw. I appreciate for the opportunity which he offered me to be part of SaToSS. Whenever I was in trouble, he was always there to help me out. Sjouke is a wonderful supervisor because of his constant trust in students. I am very grateful for his harsh but helpful comments, valuable suggestions on my writing skills and moral support.

I would like to express my special thanks to Gabriele Lenzini. I really enjoy the days we spent together in New Orleans, walking down the beautiful streets and tasting the giant American burgers. Collaborating with industry partners always comes with a lot of extra efforts. I want to thank Gabriele for his leading role in project management and the huge amount of administrative work on maintaining a good relationship with our partner. I am very grateful for all he has done to let me focus on my research.

I want to thank Carlo Harpes, Miguel Martins, Benoît Jager and all other people in itrust consulting, Luxembourg, who devoted their time and efforts to the LASP project. It was a pleasure to work with this young and energetic team and I appreciate very much for their technical support.

I thank the members of my defence committee, Yves Le Traon, Ling Liu and Catuscia Palamidessi for their time and valuable feedback. It is a great honour for me to have them in my defence committee.

I want to thank my colleagues: Naipeng Dong, Claudio Fiandrino, Barbara Kordy,

Piotr Kordy, Matthijs Melissen, Tim Muller, Saša Radomirović, Patrick Schweitzer, Rolando Trujillo Rasúa, Ton van Deursen, Qixia Yuan and Yang Zhang for their inspiring and informative discussion. My special appreciation goes to Andrzej Mizera. It was a great experience to work with him and thanks for his mathematical help on my last piece of work in my PhD period.

During my Ph.D study, I co-supervised three master students with their master theses. They are David Fonkwe, Ran Xue and Ruipeng Lu. I am very happy that they all have found good jobs and I wish them a good future.

I would like to thank my parents and sisters for their unconditional love. I also want to thank my nieces and nephews for the happiness they brought to me. My last but most sincere thanks belongs to Yining, my loved wife, who makes me the luckiest man in the world.

Xihui Chen

Contents

1 Introduction **1**

1.1 Location-based Services 2
1.2 GNSS Systems and Their Vulnerabilities 3
1.3 Location Assurance 4
1.4 Privacy in LBSs 4
1.5 Research Questions 5
1.6 Thesis Overview 7
1.7 Origins of the Material 8

I Location Assurance **11**

2 Preliminaries **13**

2.1 GNSS Signals 13
2.2 GNSS Signal Spoofing 15
2.3 Subjective Logic 16

3 A Trust Framework For Evaluating GNSS Signal Integrity **19**

3.1 Introduction 19
3.2 A Trust Framework 20
3.2.1 GNSS systems 21
3.2.2 GNSS receivers 21
3.2.3 Signal integrity 22
3.2.4 Adversary model 23
3.2.5 Spoofing detection methods 24
3.3 Deriving Validity Opinions 26
3.3.1 Distance of measurements to reference sets 27
3.3.2 Degradation function 27
3.4 Inferring Signal Integrity 28
3.4.1 Stateless spoofing detection 28

3.4.2 Stateful spoofing detection 29
3.4.3 Determining the conditional opinions 30
3.5 Combining Integrity Opinions 32
3.5.1 The Veto algorithm 33
3.5.2 The Consensus algorithm 33
3.5.3 The Combined algorithm 35
3.6 Prototyping 36
3.7 Validation 37
3.7.1 The experimental setup 38
3.7.2 Experimental results 38
3.8 A Demonstrator: Location Assurance Provider 42
3.9 Related Work 44
3.10 Conclusion 45

II Query Privacy 47

4 Protecting Query Privacy 49
4.1 Introduction 49
4.2 Our Framework 50
4.2.1 Mobile users 50
4.2.2 Request generalisation algorithms 51
4.2.3 The adversary 51
4.2.4 Classifying contextual information 53
4.3 Privacy Analysis based on User Profiles 54
4.4 Privacy Analysis based on Query Dependency 56
4.4.1 Updating the adversary's knowledge 57
4.4.2 Deriving query dependency 57
4.4.3 Query privacy analysis 59
4.4.4 Handling the time intervals between requests 61
4.5 Measuring Query Privacy 63
4.6 Generalisation Algorithms 65
4.6.1 An algorithm for k-ABS 65
4.6.2 An algorithm for α-USI and β-EBA 66
4.7 Experimental Results 71
4.7.1 Impact of contextual information 72
4.7.2 Effectiveness of the new privacy metrics 74

4.8 Related Work . . . 79
4.8.1 Query privacy and request generalisation . . . 79
4.8.2 Context-aware privacy analysis . . . 80
4.8.3 Area generalisation algorithms . . . 81
4.9 Conclusion . . . 82

III Location Privacy 83

5 Activity-targeted Location Privacy Attack 85
5.1 Introduction . . . 85
5.2 System Model . . . 87
5.3 Profiling Users . . . 89
5.3.1 Mobility profiles . . . 89
5.3.2 Request issuing patterns . . . 92
5.4 A New Tracking Attack . . . 94
5.4.1 De-anonymisation . . . 94
5.4.2 De-obfuscation . . . 97
5.4.3 Discussion . . . 98
5.5 Localisation Attack . . . 99
5.6 Validation . . . 100
5.6.1 Constructing mobility profiles . . . 100
5.6.2 Evaluating privacy attacks . . . 103
5.7 Related Work . . . 109
5.8 Conclusion . . . 110

IV Concluding Remarks 111

6 Conclusion and Future Work 113
6.1 Conclusion . . . 113
6.2 Future Work . . . 114
6.2.1 Adding behaviour perturbation . . . 114
6.2.2 Adding dependency between users . . . 115

Bibliography 117

Publications 129

Index of Subjects **131**

Curriculum Vitae **133**

List of Figures

1.1 Roles involved in location-based services. 3
1.2 Thesis overview. 7

2.1 Generation of civil signals 14

3.1 The sequential steps of a spoofing detection method. 24
3.2 An example of degradation functions and distances to reference sets 28
3.3 The components of the prototype. 36
3.4 The integrity opinions. 39
3.5 Integrity Opinions of individual detection methods. 40
3.6 Combined opinions of integrous signals. 41
3.7 Location Assurance Provider (LAP). 43
3.8 Testing result of the LAP. 44

4.1 A centralised framework of LBSs. 51
4.2 A classification of contextual information. 54
4.3 A history window of n observed requests. 59
4.4 The three cases. 61
4.5 An example execution of our algorithm uniformDP. 70
4.6 Impact of user profiles and query dependency on Δp. 73
4.7 Δp vs. #active users and n. 74
4.8 The impact of K. 75
4.9 Impact of history window size n. 76
4.10 Impact of dependency $\Pr(q_i \,|\, q_{i-1})$. 78
4.11 Average computational time (history window $n = 3$). 79

5.1 A user and his trajectory. 88
5.2 An example of $\mathcal{P}_u$. 91
5.3 The distribution of users' PoIs. 101
5.4 The log-likelihood of orders. 102

5.5 The probability density function of stay time and transition time. . 103
5.6 Estimation error vs. length of period. 106
5.7 Mean estimation error vs. length of period. 107
5.8 Mean estimation error vs. # PoIs. 108

List of Tables

3.1 Notations in Chapter 3 22
3.2 The parameters used in stateless detection. 37
3.3 Belief & uncertainty bounds of integrity opinions. 40
3.4 The average integrity opinions of integrous and spoofed signals. . . 42

4.1 Notations in Chapter 4 53
4.2 Increases of posterior probabilities and area of generalised regions. 77

5.1 Notations in Chapter 5 90
5.2 The number of users with transition matrices of different orders. . . 103
5.3 The distribution of estimation error. 107

1

Introduction

Curiosity is human nature and drives people to explore the unknown world. From the overland Silk Road to the discovery of the 'New World', the history of human being is actually a history of exploration. Thanks to the thousands of years' efforts from billions of brilliant people, our sphere of activities has been expanded to all over the globe. Meanwhile, our transportation means have also evolved from horse-driven chariots to high-speed rails and planes. The enlarged area of activities and ease of mobility have made travelling part of our daily lives. For example, in Beijing, a person spends 66 minutes a day on average commuting between home and work.

In this modern era which is featured by the explosion of information, people have become accustomed to electronic connectivity such as emails and short messages. They expect that this connectivity can be continued even when they are out of their offices and houses. Consequently, an effective method is in need to cover the gap of connectivity caused by travelling. The innovations of information technologies meet this requirement. Fast wireless networks such as 3G and 4G provide the infrastructure that enables us to connect to the Internet ubiquitously in a large speed. For instance, 4G networks have covered most of areas worldwide and even promise a peak downloading speed up to 1Gbps. Smartphones and other mobile devices provide us with tools to request and receive information anywhere and anytime. In 2012, the number of smartphone users first surpassed one billion and it is foreseen that in 2014 this number will grow by three quarters to 1.75 billion. The merge of these technologies has shown its impacts on our lives. We are used to seeing people checking emails in meetings, reading news on buses and posting messages on Facebook during lunch.

Researchers at the University of California reported that about 30% more information was generated worldwide every year between 1999 and 2002 [LV03]. We believe that this rapid growth continued in the last decade, considering the increasing number of Internet users. Facing the increasing amount of information, people desire not only ubiquitous access to information, but also fast and precise access. In other words, people need information that is catered according to their own requirements, i.e., *personalised information*. For instance, instead of a complete list of newly published papers in all areas, scientists prefer to get informed of the ones relevant to their research domains. Instead of manually specifying requirements, an enormous number of people allow information service providers to gather their personal information from which their requirements are automatically extracted. For example, we grant Google the right to collect our daily web usage history so as to obtain personalised search results and video recommendation on Youtube.

The mobility of connectivity opens up an opportunity of providing better person-

alised information services because the constant changing context during people's travelling can be added as a new source of personal information. Information services can thus be adjusted according to users' locations and nearby surroundings. In this thesis, we concentrate on one of the most popular classes of services that provide such context-aware information: *Location-based services* (LBS). An LBS is a service accessed by mobile devices, which provides information or functionalities customised according to the locations of the devices. For instance, in some cities, people can ask the question 'when will the next bus no. 7 arrive at this stop?' where 'this stop' is determined by the GPS[1] chips embedded in their smartphones. Nowadays, we have got used to checking nearby restaurants on Google and calling the nearest available taxies through mobile applications. Mobile health-care, electronic toll systems and mobile gaming, which only existed in scientific fiction 20 years ago, now have become reality.

1.1 Location-based Services

'Active Badge', developed in 1992, is recognised as the first experimental LBS [Yu08]. With an indoor positioning system tracking employees' locations, messages are forwarded to the target employees directly, instead of being broadcast in the entire building. Since then, LBSs have been implemented for a number of scenarios. Currently, social networking, e.g., finding nearby friends and geo-social networks, is the first LBS application in terms of the number of users and revenues. Following social networking are local information search (e.g., queries about nearby restaurants) and navigation services. These top three LBSs have one thing in common: a user makes use of his own locations when requesting LBSs. However, in some LBSs, users can request information based on others' whereabouts. Car rental companies are known to track their fleets so as to monitor whether their clients violate rental agreements, e.g., by driving over the speed limit [Goo09]. In this thesis, we concentrate on the former class of LBSs where users request LBSs using their own locations due to their large number of users.

We recognise three roles in an LBS scenario: *location provider*, *user*, *LBS provider*. A location provider is a system that either provides necessary information to calculate locations or directly calculates the locations for mobile devices. A user carries a mobile device and the locations calculated by location providers actually correspond to his movements. Whenever in need of an LBS, the user specifies the information required as a *query* and sends it with his location to an LBS provider. An LBS provider implements a software system taking location owners' locations as input and outputting the response catered according to LBS requesters' queries. In the example of requesting the arrival time of the next bus, the bus monitoring centre is the LBS provider while GPS plays the role of the location provider. People are the users and the query is the arrival time of the next bus. Figure 1.1 depicts the roles as well as steps involved to receive an LBS and the information exchanged in such LBSs.

From Figure 1.1, we can see that the location provider is essential for LBSs. It is where users' locations originate and constructs the first step to obtain an LBS.

[1]GPS is short for the American global positioning system.

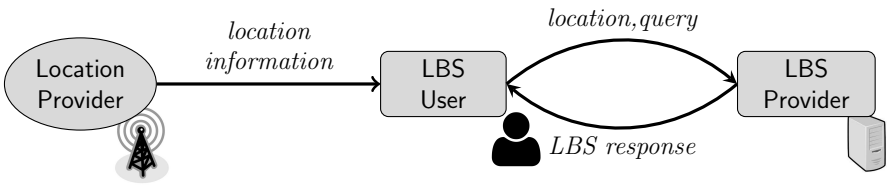


Figure 1.1: Roles involved in location-based services.

In practice, positioning systems play the role of location providers, so whether they can do their job as expected determines the trustworthiness of locations. LBSs stem from the launch of global navigation satellite systems (GNSS) . Due to the free and ubiquitous positioning services, GNSS have been widely deployed in terms of the number of users and application scenarios. The European GNSS Agency predicted that in 2022, seven billion devices will be equipped with the access to GNSS [Age13]. In the next section, we briefly describe GNSS and their vulnerabilities.

1.2 GNSS Systems and Their Vulnerabilities

GNSS systems. As the first GNSS system, the American *global positioning system* (GPS) is designed to determine positions on the surface of the earth with accuracy of meters or even centimetres. It was initially developed only for military use. Since 1990s, the full operational service has become freely accessible to civilian users but with a downgraded position accuracy . A location is calculated based on comparing the timing of signals transmitted from the set of satellites above the horizon. GPS is not the only operational GNSS system. Russia has launched a comparable system, GLONASS, and the system that became most recently operational is the Chinese Beidou system. The European Galileo system will become operational in a few years.

In spite of the high accuracy with which locations are determined, GNSS cannot guarantee constant availability. It is estimated that in Hongkong, due to the influence of urban canyons, GPS signals are only available for 20% of the time. The reason is that GNSS signals are transmitted in a relatively low power level and over great distances. The received GPS signals are weak and easily further attenuated by by walls and roofs. Since the reception of GPS signals is not reliable indoor or in other obstructed environments, complementary positioning systems are developed, e.g., *cell identification*, *Wifi fingerprinting* and RFID. A detailed summary of other positioning techniques can be found in [Pri13].

Attacks on GNSS systems. As we mentioned, GNSS signals for civilian purposes have a weak signal strength Different from military signals, GNSS civil signals are not encrypted or signed. Thus, there is no way to authenticate where they are originated. Because civil signals are broadcast in the open air, they are easily interfered with or even taken over by false signals. Such attacks are called *jamming* and *spoofing* [WJ03, WHD+05, Bor07, POB+13], respectively. Jamming causes loss of meaningful GNSS signals and prevents receivers from calculating a valid location. Spoofing is more sophisticated. Its purpose is to fool a receiver to calculate

another location which may be far from where it is through feeding it with faked signals. The vulnerability of GNSS to spoofing has been proved in theory and illustrated by real-life experiments. In June 2013, students from the University of Texas successfully controlled a superyacht and piloted it off its planned course by subverting GPS signals without being detected by the onboard navigation system.

1.3 Location Assurance

In conventional services, a user requests a service with a purpose and always expects the service provider to return a service that meets this purpose. For instance, we go to a restaurant with the expectation that the restaurant will offer us a good meal. However, with respect to LBSs, whether a user's goal can be satisfied not only depends on LBS providers but also the locations sent to LBS providers. Consider that a wrong location is sent to Google when we query the nearest gas stations. We may drive hundreds of extra miles before getting refuelled in spite of Google's reliable service and the good coverage of petrol stations. In some cases, vulnerable locations may even result in irreversible losses, e.g., people's lives and even homeland security. For instance, as in more than 50% of emergency calls, the callers cannot provide a valid location, the enhanced 911 service in America requires cell phone operators to provide callers' precise locations. In this enhanced emergency service, if a location is miscalculated, the loss may be a person's life.

In LBSs, we rarely input locations manually but instead rely on other systems to obtain locations, e.g., GNSS systems and cell phone operators. On the one hand, manual input is cumbersome, especially during fast movement. On the other hand, locations are usually not known by users, e.g., when users drive on highways. As a result, the location calculation is not under users' control. Considering the vulnerability of GNSS systems to spoofing attacks, in order to ensure the quality of service and even security, users need an assurance that the locations are correct and precise. We call this requirement *location assurance*.

1.4 Privacy in LBSs

Privacy is a human right and should be respected whenever users interact with electronic systems. LBSs are not exceptions. When making use of LBSs, users expose their locations and queries. Both of them can be explored by attackers to infer users' private information. Such malicious inference in turn threatens users' privacy in LBSs. First, locations can serve as a piece of subsidiary information to peek users' personal life. For instance, hospitals are public places and the location of a hospital itself does not carry any sensitive information about users. However, it will become sensitive when the functionality of hospitals and the purpose of people in hospitals are taken into account. An appearance in a cancer centre reveals that a person may suffer from a bad health problem. In order to avoid the abuse of inferred personal information, users desire the protection of *location privacy* in LBSs. Second, even if where users are located does not reveal any sensitive information, their queries may still put their privacy at risk. By 'query', we mean the specification of the information or functionality a user wants to acquire from

an LBS. For example, a query for nearby casinos will reveal a user as a fan of gambling which is usually not accepted as a healthy hobby. The potential leakage of privacy by queries leads to users' requirement for *query privacy*.

The requirements of LBS users discussed above form the focus as well as the title of this thesis:

Location assurance & privacy in location-based services.

1.5 Research Questions

Currently GNSS systems are the most used positioning systems. As we have mentioned, they are vulnerable to spoofing attacks which can result in miscalculated locations. Therefore, assurance for trustworthy locations cannot be guaranteed if locations are calculated with GNSS systems. To eliminate spoofing, we need to change GNSS infrastructures so that signals can be authenticated. This cannot be achieved in a short term due to the high cost. Therefore, from users' perspective, they want to learn whether their locations are trustworthy. If not, to what extent they can trust their locations. With such information, they can decide whether to request LBSs. In such situations, even if the quality of LBSs is degraded, users can be informed and get prepared in advance for the degradation.

To calculate a GNSS location, GNSS receivers capture the GNSS signals that arrive at them. Then based on the information carried by these signals, they accomplish the calculation. Thus, whether the calculated location is trustworthy is first determined by the quality of received GNSS signals. If we can ensure that the received GNSS signals originated from GNSS satellites and that they have not been modified artificially, then with a well-protected receiver, we can ensure the assurance of locations. To achieve this goal, we have to answer the following research question:

Research question 1: *Can we assess the integrity of GNSS signals?*

In the last decade, privacy in LBSs has raised people's concern and many methods have been proposed or deployed to protect users' privacy. The essence to protect the privacy of a user is to hide the user's association to the information that he wants to keep secret. The first approach to protect users' privacy is to encourage users not to reveal information when interacting with outside environments. This does not work for personalised services such as LBSs as personal information is a necessary input. In such cases, we can turn to the second approach which implements legal means to protect personal information from being misused. For instance, U.S. legislation has regulated the circumstances when cellphone operators can release users' locations since 1999. However, legal actions can only be taken after users' privacy is violated. They are ineffective to impede the violation. The third mechanism is to resort to privacy enhancing technologies (PET) which help users reveal less private information. PETs can provide protection before any privacy is breached so they can naturally complement the privacy protection given by legal actions. In this thesis, we focus on PETs and study their insufficiencies and possible improvements.

The PETs in LBSs can be divided into two classes: *cryptography-based* and *non-*

cryptography-based. The former class explores cryptographic methods to encrypt requests so as to hide users' locations and queries from attackers as well as LBS providers [KSSM11]. Non-cryptography-based techniques aim to protect user's privacy by modifying the information contained in LBS requests before they are sent to LBS providers. A straightforward method of this class is to replace users' identities with pseudonyms [BHV07, CPHL07, FMHP09], called *anonymisation*. However, this method has been shown to be insufficient for both query and location privacy since the time-stamped locations contained in LBS requests can serve as quasi-identifiers sometimes. For instance, in some scenarios, users' locations can be obtained through other ways, e.g., camera surveillance. Besides, as users possess certain daily routines, their locations at certain time points can be inferred. For example, users normally stay in offices during working hours. Then based on the location in an LBS request, the issuer can be identified. In such cases, *obfuscating* methods can be used. Users can replace their locations in requests with larger areas [GG03, XKP09], which is called area cloaking (also called *generalisation*). In addition, they can issue dummy requests or choose to hide certain requests.

With the development of information technologies and research, new privacy risks will emerge. With respect to query privacy, the adversary can improve the inference of query issuers when she has access to additional information about users. Consider that a query about nearby beauty saloons is issued from an area. The query issuer is a professional lady. She believes that she is well protected since all users in the area have the same chance to be identified as the issuer. However, other users are all men. If the adversary explores users' genders, the lady will be more likely to be distinguished. This example shows that *contextual information* (e.g., genders and professions) can be exploited and threaten users' query privacy. Although various types of contextual information have been identified and analysed [MBFW07, SAV08, SAV11, CZBP06, RPBJ09], they are studied independently and the privacy protection methods proposed are only effective for the identified contextual information. As a consequence, we need a unified method which is able to analyse distinctive types of contextual information and a general mechanism to protect users' query privacy. This leads to our second research question:

Research question 2: *How can we protect users' query privacy against the adversary with various contextual information?*

With respect to location privacy, many works in the literature have shown that users can still be associated with their locations [STBH11] even if location privacy preserving methods are used. While existing attacks on location privacy in the literature mostly target at deriving 'where users actually went', recent research requires us to revisit this objective from the view of practical attackers. Namely, what the adversary is really curious about with respect to location privacy is what activities users did during their movement [LOYK11]. For instance, receiving medical treatments reveals a user's poor health condition more precisely than just a visit to a hospital. Therefore, users are exposed to a new threat to location privacy which targets at their activities. Our last research question is related to this threat:

Research question 3: *How can we formally capture the threat to users' location privacy which targets at users' activities?*

1.6 Thesis Overview

This thesis is structured into three parts each of which addresses one research question as shown in Figure 1.2. In Part I, we address Research question 1 and develop a framework to evaluate the trust a user can put in the integrity of received GNSS signals. In Part II, we answer Research question 2. We develop a probabilistic framework and formally define the attack on users' query privacy which an adversary possessing various types of contextual information can make. In Part III, we state Research question 3 and study how to attack on location privacy to derive users' activities.

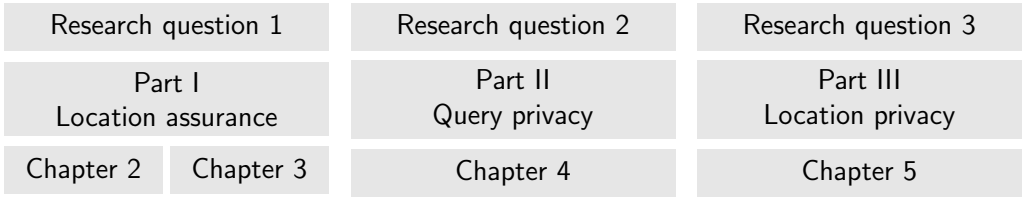


Figure 1.2: Thesis overview.

The contributions of each chapter can be summarised as follows:

- **Chapter 2: Preliminaries**

  We present the preliminary knowledge about GNSS signals and positioning algorithms. We also give a short introduction to a probabilistic logic, *subjective logic*, which can be used to conduct probabilistic reasoning with uncertainty or incomplete evidences.

- **Chapter 3: A trust framework for evaluating GNSS signal integrity**

  We develop a novel trust framework based on subjective logic to evaluate the integrity of received GNSS civil signals. We are the first to formalise *signal integrity* and use it to classify existing spoofing detection methods. All such methods make use of evidences gathered from received signals to derive a conclusion on signal integrity. We identify the right reasoning that should be used to draw such conclusions from evidences. Originally, we measure the uncertainty that comes with the reasoning and is insufficiently studied in the literature of evaluating signal integrity. Our framework also gives rise to several natural ways to combine the conclusions given by various spoofing detection methods to reach an overall evaluation of signal integrity. We implement a prototype of our framework and show its effectiveness through experiments on both real and simulated spoofed signals. In order to show its potential marketing values, we implement a public service called *location assurance certification*.

- **Chapter 4: Protecting query privacy in location-based services**

  We classify contextual information related to LBS query privacy and focus on two types of contextual information: *user profiles* and *query dependency*. User profiles have been insufficiently studied in LBS query privacy protection, while we are the first to show the impact of query dependency on users'

query privacy. Specifically, we present a general framework to enable the attackers to compute a distribution on users with respect to issuing an observed request. The framework can model attackers with different contextual information. We take user profiles and query dependency as examples to illustrate the implementation of the framework and their impact on users' query privacy. With our framework, we show the insufficiency of existing query privacy metrics, e.g., k-anonymity, and propose several new metrics. We also develop new generalisation algorithms to compute regions satisfying users' privacy requirements expressed in these metrics. By extensive experiments, we show that our metrics and algorithms are effective and efficient for practical usage.

- **Chapter 5: Activity-targeted location privacy attack**

  We propose a new attack to breach users' *activity privacy*. Specifically, our attack uncovers the places where a user performed activities and reveals the starting and ending time of each of his activities. Compared to existing methods to derive such information, our attack takes as input the exposed locations, protected by location privacy preserving methods. To perform this attack, we propose a new model to capture users' mobility and usage of LBSs in continuous time, which naturally expresses users' behaviour patterns in LBSs and ensures the precision of the calculated temporal information. We then adopt and extend an existing framework for quantifying location privacy and formally implement our attack. Through experiments on a real-life dataset, we show that our new tracking attack is quite effective. In addition, in order to show the generality of our model for user profiles, we also implement the common attacks on location privacy and evaluate them through experiments.

1.7 Origins of the Material

Chapter 3 summarises the achievements in a European Space Agency (ESA) project "Developing a Prototype of Localisation Assurance Service Provider (LASP)". This is a collaborative project between itrust consulting s. à r. l., Luxembourg and SnT (Interdisciplinary Centre for Security, Reliability and Trust) of the University of Luxembourg. The theory about our trust framework is based on a joint paper with Gabriele Lenzini, Miguel Martins, Sjouke Mauw and Jun Pang [CLM+13]. The implementation of our prototype and the demonstration of LASP is based on the papers [HMC+12, CHL+13a, CHL+13b]. A Luxembourgish patent [CCL12] is also approved based on our framework and prototype. Chapter 4 is based on a series of papers [CP12, CP13, CP14] co-authored with Jun Pang. User profiles are studied in [CP12] while query dependency is first explored in [CP13]. The probabilistic framework is proposed and validated in the extended journal paper [CP14]. Chapter 5 is based on a technical report with Andrzej Mizera and Jun Pang [CMP14].

There are a few works that are related to the general topic of this thesis but not included. In order to study location privacy in real application scenarios, I made a case study with respect to electronic toll pricing (ETP) systems. With Gabriele

Lenzini, Sjouke Mauw and Jun Pang, I proposed a new electronic toll pricing system based on group signatures, GroupETP, which achieves a good balance between privacy and overhead imposed upon user devices [CLMP12, CLMP13]. The design of GroupETP enables us to identify another risk with respect to users' location privacy. Namely, in ETP systems where location records are even anonymously stored, service providers are still able to trace users. This led to a joint work with David Fonkwe and Jun Pang [CFP12]. Based on user toll payment information, we propose a post-hoc analysis of user traceability, which aims at computing a user's all possible traces.

The collection of enormous location records in LBSs brings about new types of LBSs. Inspired by the research on user mobility profiles, together with Ran Xue and Jun Pang, I proposed a new method to construct and compare users' mobility profiles [CPX13]. This work can be explored to implement a recommender system suggesting potential friends based on their movement similarity. This piece of work was later extended with location and temporal semantics and led to a journal paper [CPX14]. Collaborating with Ruipeng Lu and Jun Pang, I extracted and formalised the basic principles that should hold in calculating users' similarity based on trajectory patterns [CLMP14]. This piece of work also led to a paper [CKLP14] demonstrating a tool which provides an integrated interface to construct and compare users' trajectory patterns. This tool can be downloaded from `http//satoss.uni.lu/software/MinUS/`.

Part I

Location Assurance

2

Preliminaries

To fully comprehend the following chapter, some preliminary knowledge is required. We briefly introduce it in this chapter. We first describe what composes GNSS (Global Navigation Satellite Systems) signals and how to calculate a GNSS position. We recall subjective logic [Jøs12] and explain the basic concepts of *opinions* and *conditional reasoning*. In addition, we present the implementation of the *consensus* operator on subjective logic opinions.

2.1 GNSS Signals

GNSS signals from a GNSS system are broadcast to the earth by a constellation of satellites. The most famous GNSS system is GPS (Global Positioning System) which we take as a representative in this thesis. Other systems, such as GLONASS and Galileo, have similar structures.

GPS satellites are equipped with atomic clocks which are synchronised with the universal time. GPS signals are transmitted in the open air in two frequencies f_{L1} and f_{L2}. GPS signals are composed of the following three parts [Bor07]:

- *Carrier.* There are two carrier waves with frequency f_{L1} and f_{L2}, respectively.
- *Navigation data.* Navigation data carry information about the orbits of satellites. Such information is uploaded to all satellites from the ground stations. The navigation data are transmitted in a bit rate of 50 bps.
- *Spreading codes.* A spreading code is actually a sequence of bits with a fixed length. Each satellite has two spreading codes: the coarse acquisition (C/A) and the encrypted precision code (P(Y)). A C/A code is a sequence of 1023 bits transmitted in a rate of 1.023 MHz while a P(Y) code is a longer sequence with 2.35×10^4 bits and transmitted in a rate of 10.23 MHz. This means that the C/A code repeats itself every millisecond and that the P(Y) code repeats itself every week. The C/A code is publicly known and encoded in civil signals while the P(Y) code is encrypted and can only be accessed by certified military devices.

A satellite generates its signals by modulating its spreading code and navigation data with the carrier waves. Note that C/A codes are only modulated with the carrier with frequency f_{L1} while P(Y) codes are modulated with both frequencies. As we focus on civil applications of GPS systems, throughout the thesis we only

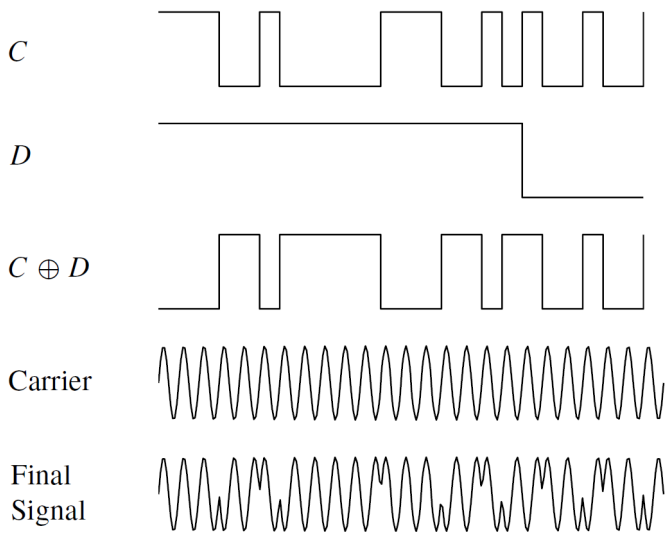


Figure 2.1: Generation of civil signals [Bor07].

consider scenarios where civil signals are used. Thus, we simply refer to civil signals as signals. In Figure 2.1, we show the generalisation of the civil signals of a satellite. The C/A code (C) is first combined with the navigation data (D) based on the logic operation of *exclusive or* ($\oplus$). This combination generates a sequence of bits $C \oplus D$. After modulating this sequence with the carrier wave, we have the final signals which will be transmitted from the satellite to the earth.

A GPS receiver antenna captures signals from the satellites which are above the horizon. From these signals, the receiver calculates a three-dimensional coordinate as follows. The receiver simulates the bit stream of each satellite's C/A code. At any time point, any simulated stream outputs the same bit as the one generated by the corresponding satellite if the clock on the receiver is perfectly synchronised with the clock on the satellite. In other words, a receiver has the replicas of the C/A codes of all the deployed satellites. Based on the C/A codes, the receiver separates the signals originated from different satellites and measures their time offsets with the replicas. Such an offset is in fact equivalent to the amount of time which a received signal has taken to reach the receiver from the originating satellite. By multiplying the time offsets with the speed of light, we can thus obtain the distances between the receiver and the satellites in range. These distances are called the *pseudoranges* of the satellites with respect to the receiver. Let ρ be the pseudorange of a satellite in range whose position is (x, y, z). Consider that a user is located at the position (x_u, y_u, z_u). Since a pseudorange actually measures the length of the line segment between a satellite and the receiver, we have the following equation:

$$\rho = \sqrt{(x - x_u)^2 + (y - y_u)^2 + (z - z_u)^2}. \tag{2.1}$$

As the navigation data includes the satellite's location, i.e., (x, y, z), we have only three variables to solve. Thus with three satellites, we can compute a three-dimensional location in theory. However, in practice, the local clock of a receiver cannot be perfectly synchronised with the clocks on satellites. When such time

differences are taken into account, the above equation can be reformulated as:

$$\rho = \sqrt{(x - x_u)^2 + (y - y_u)^2 + (z - z_u)^2} + c \cdot \delta t \tag{2.2}$$

where δt is the time difference between the local clock of the receiver (located at (x_u, y_u, z_u) and the atomic clock on the satellite (located at (x, y, z)). Due to the correcting scheme of the GPS ground-monitoring network [KH05], the clocks on satellites can be considered to be synchronised with the universal time. Thus, all satellites share the same time difference with respect to a receiver which adds a new unknown variable to the calculation of GPS locations. A fourth satellite is thus required.

2.2 GNSS Signal Spoofing

Intuitively, the main idea of signal spoofing is to feed a GNSS receiver with false signals that can fool it to calculate a different location from where it is. According to the localisation mechanism of GNSS systems in Equation 2.2, the calculation of (x_u, y_u, z_u) depends on the parameters ρ and (x, y, z). Essentially, signal spoofing interferes with the calculation of locations by changing the values of these parameters. Two ways to implement signal spoofing have been identified in the literature.

(i) **Manipulating pseudoranges.** Because C/A codes are public and no authentication mechanisms exist to protect them, an attacker can construct a signal containing a C/A code with arbitrary time offset to the one simulated by a receiver. This forgery will lead the receiver to calculate an incorrect pseudorange of a satellite.

(ii) **Manipulating navigation data.** Since the format of navigation data is also publicly known, an attacker can generate navigation data with arbitrary information but conforming with the format. In this way, the receiver will learn an incorrect location of the satellite.

By either or both of these two ways, a receiver can be fooled to calculate any location, no matter where it is actually.

The above two ways of spoofing have been validated in the literature. Using the first approach, Humphreys et al. [HLP+08] implement a simulator which uses a GPS receiver to decode GPS signals and then broadcasts them with arbitrary delays. Tippenhauer et al. [TPRC11] theoretically prove that an attacker can spoof multiple receivers at the same time by carefully deploying broadcasting antennas in certain positions. These positions simulate the geometry of satellites. With respect to the second approach, Nighswander et al. [NLD+12] implement a simulator which re-broadcasts signals with arbitrary navigation messages. This method can attack multiple receivers more efficiently in larger areas compared with the simulator of Tippenhauer et al. [TPRC11] as satellites' geometry is ignored.

2.3 Subjective Logic

Subjective logic was first introduced by Audun Jøsang [Jøs12]. It is a probabilistic algebra which combines belief and disbelief about statements and keeps track of uncertainty during reasoning.

Subjective logic opinions. In subjective logic, an *opinion* expresses the belief about one or multiple propositions from a space called the *frame of discernment*. An opinion over a frame X is a composite function consisting of three components: a belief function, an uncertainty mass and a base rate function. The belief function assigns belief mass to each proposition in X, which can be interpreted as the positive belief on the truth of the element. It is sub-additive, meaning that the sum of all propositions' belief mass is not larger than 1. Uncertainty mass is the amount of belief that is not assigned as belief mass. It is represented by the perceived imprecision of the probability estimates. The base rate function expresses the *a priori* probability of each proposition in X being true.

Definition 2.1 (Subjective logic opinion)**.** *Let X be a frame $\{x_1, \ldots, x_n\}$. An opinion on X can be represented by $w_X = (\vec{b}_X, u_X, \vec{a}_X)$ where $\vec{b}_X : X \to [0,1]$ is the belief function, $u_X \in [0,1]$ is the uncertainty mass and $\vec{a} : X \to [0,1]$ is the base rate function. Furthermore,*

$$\sum_{x \in X} \vec{b}_X(x) \leq 1; \quad u_X = 1 - \sum_{x \in X} \vec{b}_X(x); \quad \sum_{x \in X} \vec{a}_X(x) = 1.$$

The expectation probability of $x \in X$ being true is:

$$\vec{E}_X(x) = \vec{b}_X(x) + \vec{a}_X(x) \cdot u_X. \tag{2.3}$$

Consider a binomial frame X denoted by $\{x, \bar{x}\}$ where $\bar{x}$ is the negation of x. Then the opinion about the truth of x can be denoted as $w_x = (b, d, u, a)$ where $b = \vec{b}_X(x)$, $d = \vec{b}_X(\bar{x})$, $u = u_X$ and $a = \vec{a}_X(x)$ indicating the belief, disbelief, uncertainty and the *a priori* rate about x being true. The expectation probability of x being true is $E(w_x) = b + a \cdot u$.

Conditional belief reasoning. Conditional reasoning has been discussed in both binary logic and probability calculus. It offers a way to calculate the truth of a proposition y based on the evidence about another proposition x which has a conditional relation with y.

According to the causal relation, we have *deductive* reasoning and *abductive* reasoning. If x is the antecedent, then the reasoning is deductive. If y is the antecedent, then the reasoning is abductive. Compared to the probabilistic method, subjective logic takes opinions as input in the reasoning and thus captures the underlying uncertainty.

Deduction and abduction on binomial frames, i.e., $X = \{x, \bar{x}\}$ and $Y = \{y, \bar{y}\}$

have the following notations:

$w_{y|x}$: conditional opinion on y given x being TRUE;
$w_{y|\bar{x}}$: conditional opinion on y given x being FALSE;
w_x : opinion on the proposition x;
$w_{y\|x}$: opinion on y deduced/abduced from the observation on x.

Assume we have a causal conditional between x and y, i.e., "if x then y" (denoted by $x \to y$) and $w_{y|x}$ and $w_{y|\bar{x}}$ are learned. If we have an observation on x which gives the opinion w_x, then the deduced opinion on y should be calculated by considering both of the situations when x is TRUE and FALSE. In subjective logic, '$\circledcirc$' is used as the operator calculating the opinion on y given w_x and the two conditional opinions $w_{y|x}$ and $w_{y|\bar{x}}$, i.e., $w_{y\|x} = w_x \circledcirc (w_{y|x}, w_{y|\bar{x}})$. If we have evidence on y i.e., the opinion w_y, then the opinion on x can be calculated by abductive reasoning. The idea is to calculate $w_{x|y}$ and $w_{x|\bar{y}}$ based on $w_{y|x}$ and $w_{y|\bar{x}}$ using the Bayesian theorem, where the *a priori* probability of x, i.e., a_x, is required. In this way, deductive reasoning can thus be used. In subjective logic, $\overline{\circledcirc}$ is the abductive operator calculating w_x based on $w_{y|x}$, $w_{y|\bar{x}}$ and a_x, i.e., $w_{x\|y} = w_y \overline{\circledcirc}(w_{y|x}, w_{y|\bar{x}}, a_x)$. We refer the readers to [Jøs09, JPD05] for the details of the implementation of the operators.

Conditional reasoning is applicable on multinomial opinions as well. Suppose two multinomial frames X and Y. Assume conditional opinions $w_{Y|X}$ and $w_{Y|\overline{X}}$ are available. Note that $w_{Y|X} = \{w_{Y|x} \mid x \in X\}$ and $w_{Y|\overline{X}} = \{w_{Y|\bar{x}} \mid x \in X\}$ where $w_{Y|x}$ (resp., $w_{Y|\bar{x}}$) represents the conditional opinion on Y given that x is TRUE (resp., FALSE). The opinion on Y based on observations on X (i.e., w_X) can be calculated by deductive reasoning, i.e., $w_{Y\|X} = w_X \circledcirc w_{Y|X}$. Likewise, the opinion on X based on observations on Y can be calculated by abductive reasoning, i.e., $w_{X\|Y} = w_Y \overline{\circledcirc}(w_{Y|X}, \vec{a}_X)$ where $\vec{a}_X$ is the *a priori* distribution on X.

Consensus operator ($\oplus$). Given a proposition, when many sources of evidences are available, each of them will lead to an opinion on the proposition. Therefore, a method is required to combine such opinions so as to reach an overall conclusion about the proposition. When evidences are independent of each other, for instance, readings obtained of a patient's body temperature in non-overlapping periods, we can make use of the *consensus* operator (also called cumulative fusion) to combine subjective logic opinions.

Given two subjective opinions, the consensus operator is defined as follows:

Definition 2.2 (Consensus $\oplus$)**.** *Let w^A and w^B be opinions respectively held by agents A and B over the same frame $X = \{x_j \mid j = 1, \ldots, l\}$. Let $w^{A\diamond B}$ be the opinion such that*

Case I: For $u^A \neq 0 \vee u^B \neq 0$:

$$b^{A\diamond B}(x_j) = \frac{b^A(x_j)u^B + b^B(x_j)u^A}{u^A + u^B - u^A u^B} \tag{2.4}$$

$$u^{A\diamond B} = \frac{u^A u^B}{u^A + u^B - u^A u^B} \tag{2.5}$$

Case II: For $u^A = 0 \wedge u^B = 0$*:*

$$b^{A\diamond B}(x_j) = \gamma^A b^A(x_j) + \gamma^B b^B(x_j) \tag{2.6}$$

$$u^{A\diamond B} = 0 \tag{2.7}$$

where

$$\gamma^A = \lim_{u^A \to 0, u^B \to 0} \frac{u^B}{u^A + u^B}, \ \gamma^B = \lim_{u^A \to 0, u^A \to 0} \frac{u^B}{u^A + u^B}.$$

Then $w^{A\diamond B}$ *is called the consensus opinion of* w^A *and* w^B*, representing the combination of independent opinions of* A *and* B*. By using the symbol '*$\oplus$*' to designate this belief operator, we define* $w^{A\diamond B} \equiv w^A \oplus w^B$*.*

The consensus operator satisfies the algebraic properties of associativity and commutativity [JDR10].

3

A Trust Framework For Evaluating GNSS Signal Integrity

In the previous chapter, we have learnt that GNSS signals can be forged and sent back to GNSS receivers. There is no mechanism implemented on a receiver that allows it to distinguish false signals from authentic ones. Thus, when a receiver calculates a location, it does not know whether the location is correct or not. As a consequence, users will be uncertain of the correctness of their locations even if they have kept their mobile devices well protected from virus and malwares. Users thus require a method to learn how much trust they can put in their locations.

In this chapter, we propose a trust framework to meet this requirement by evaluating the integrity of received GNSS signals. By 'integrity', we refer to the property that GNSS signals do not suffer from any artificial interference. The framework gives a formal understanding of existing spoofing detection techniques and captures the nature of uncertainty in evaluating signal integrity. With this framework, we propose three algorithms to combine the outputs from different spoofing detection methods, which may conflict with each other, into an overall evaluation. From this evaluation, users subsequently derive the assurance levels for their locations.

3.1 Introduction

It was noted by the Volpe report [Car03] in 2003 that there were no practical mitigation methods for spoofing attacks and we believe that it is still the case now, especially for GNSS civil signals. Navigation message authentication is considered as an effective method to prevent spoofing [Kuh04]. However, due to the long deployment cycle and high costs, this is not a feasible approach in the near future [TPRC11]. Instead, researchers have proposed many methods with the aim to *detect* but not to *prevent* spoofing. The general idea is to make use of some observable features that should be present when signals are not spoofed. A spoofing attack is detected if one or more of such features are not observed. For instance, under normal circumstances, the strength of GPS signals is rarely above -153.5 dBW. If a received GPS signal has a higher strength, then a detection method claims that the integrity of the signal is not preserved.

Although researchers have shown the effectiveness of their (own) detection methods through various ways, we find that the existing spoofing detection methods still suffer from the following problems:

1. The notion of signal integrity has not been formally defined, which leads

to ambiguous interpretations. Tippenhauer et al. [TPRC11] define spoofing from the viewpoint of localisation results, i.e., whether a receiver calculates the real location and time. However, this is not completely correct from the perspective of GNSS signals. In some sophisticated spoofing, the attackers may gradually fool receivers to calculate the planned position and then allow receivers to calculate the right location and time at the beginning of the attack [TPRC11].

2. Spoofing detection methods have not been systematically characterised. This leads to incorrect inference of signal integrity from the consistency of measurements with the predicted values. For example, in the inertial test [PJ08] locations cannot be correctly predicted once the past ones are calculated based on spoofed signals. In such cases, the consistency of current calculated locations does not indicate the integrity of signals.

3. The output of a detection method is always *qualitative*, i.e., whether a signal's integrity is preserved or not, while we believe that it should be *quantitative* by its nature. On the one hand, the noise from the environment always influences the receipt of GNSS signals and causes changes on certain attributes. The inconsistency of these attributes does not always come with spoofed signals. On the other hand, a powerful attacker can generate signals with certain attributes consistent with the prediction. Thus, the consistency of such attributes should not always lead to the conclusion of the signal being integrous. As we are not certain about the impacts of noise and the ability of the attackers on tuning signals' attributes, uncertainty in spoofing detection is inherently inevitable and should be quantified.

4. The outputs from different spoofing detection methods might conflict with each other and so far there exist no algorithms to combine the outputs of different methods into a coherent conclusion. Combining the results of multiple detection methods is necessary due to the fact that more evidences usually lead to more reliable conclusions.

We propose a novel trust framework based on subjective logic to evaluate the integrity of GNSS signals and address the above identified research questions. The main reasons for us to use subjective logic are that it quantifies uncertainty in logic reasoning and provides a series of operators which correspond to logic operators and take uncertainty into account.

3.2 A Trust Framework

In this section, we propose a trust framework to evaluate signal integrity. First, we precisely formulate the elements of our framework: *GNSS systems* and *GNSS receivers*, and subsequently define *signal integrity* and the *threat model*. At last we formally characterise the essential steps of spoofing detection methods.

3.2.1 GNSS systems

A GNSS system consists of a number of satellites which move in certain orbits. We denote by $\mathcal{S}$ the set of running satellites of the GNSS system. Let $\mathcal{L}$ be the set of all geographic coordinates and $\mathcal{T}$ be the set of time points. The formats of locations and time points are out of our discussion since different formats can be converted from one to another. For instance, the coordinate N25°07.450' is represented in degrees and minutes while it can also be of the form of only degrees, i.e., 25.124167. We use $\xi(S,t) \in \mathcal{L}$ to denote the real location of satellite $S \in \mathcal{S}$ at a given time $t \in \mathcal{T}$.

Satellites broadcast radio signals to the earth. GNSS signals are generated by a fixed procedure such that they have a common pattern. We take GPS signals as an example. A GPS signal includes at least two components: (1) the C/A codes of a deployed satellite (2) a navigation message with ephemeris information. Let Θ be the set of all possible GNSS signals that conform with the pattern. Note that Θ not only contains the signals that can be generated by real GPS satellites but also involves the signals that will never be transmitted by any satellites (e.g., the signals faked by attackers). In other words, as long as the signals can be correctly parsed by any GPS receivers, they are part of Θ. We use the function $sig : \mathcal{S} \times \mathcal{T} \to \Theta$ to return the signal transmitted by a satellite at a given time.

Natural factors, such as ionospheric scintillation and tropospheric effects, can attenuate signals. Attenuation can cause effects on many attributes of a signal, e.g., carrier phase advance and power decrease. Its impact is determined by the routes that signals take to arrive on the ground. As these routes are subsequently determined by where they reach and when they are generated, we use $\eta(S,\ell,t)$ to denote the attenuation on the signal of $S \in \mathcal{S}$ which is generated at time t and arrives at ℓ. We denote by $\eta(S,\ell,t) \diamond sig(S,t)$ the signal when $sig(S,t)$ reaches the earth. The signal is still an element of Θ as long as the spreading codes and the navigation data are available.

3.2.2 GNSS receivers

A GNSS receiver is a device to capture GNSS signals and calculate a location with a localisation algorithm. Recall that all GNSS signals are transmitted in the same open air with the same frequency (i.e., f_{L_1}). Therefore, the signals of all visible satellite at a given location on earth can be interpreted to be combined as a single signal. In fact, a receiver captures such a combination of the signals of all satellites in range. Let $\uplus$ be the combination operation on any two signals with the same radio frequency which is symmetric and associate. Then we can construct the set of all possible combined signals and denote it by $\mathcal{G}$. For any $s \in \mathcal{G}$, there exists a set of GNSS signals $\Theta' \subseteq \Theta$ such that $s = \uplus_{sig' \in \Theta'} sig'$. The set $\mathcal{G}$ is closed under the signal combination operation. We use $s(\ell,t) \in \mathcal{G}$ to denote the combined signal received by the receiver located at $\ell \in \mathcal{L}$ at time $t \in \mathcal{T}$.

Given a received signal, the receiver separates the GNSS signals modulated in it based on their unique features, e.g., C/A codes. This separation process can be modelled by function $sigCom : \mathcal{G} \to 2^{\Theta}$ mapping a received signal to the set of combined GNSS signals. Note that if two signals with the same C/A code are

Table 3.1: The important notations.

$\mathcal{S}$	set of running satellites of the target GNSS system
$\mathcal{T}$	set of time points
$\xi(S,t)$	position of satellite S at time t
Θ	set of possible GNSS signals
$sig(S,t)$	GNSS signal transmitted by satellite S at time t
$\eta(S,\ell,t)$	attenuation of the signal leaving S at t to reach ℓ
$\mathcal{G}$	set of combined GNSS signals that can be captured
$sigCom(s)$	set of GNSS signals combined in $s \in \mathcal{G}$
$ori(sig)$	satellite whose C/A code is modulated in sig
$loc(s)$	location and time calculated based on the received signal s
c	speed of the light
$\mathcal{I}_{s(\ell,t)}$	proposition that $s(\ell,t)$ preserves the property of integrity
$Attr(s(\ell,t))$	set of attributes of $s(\ell,t)$
$m_\alpha(s(\ell,t))$	value of attribute $\alpha \in Attr(s(\ell,t))$ of $s(\ell,t)$
$dom(\alpha)$	domain of the attribute α
$\mathcal{R}_\alpha(s(\ell,t))$	reference set of attribute α of signal $s(\ell,t)$

captured, only the one with larger signal strength is used. In this way, a GNSS receiver only takes a set of signals with unique C/A codes into further positioning calculation.

As a receiver has access to the C/A codes of all satellites, given a GNSS signal in Θ, it can identify the satellite whose C/A code is modulated. We call the satellite the *originator* of the signal. We use function $ori : \Theta \to \mathcal{S}$ to return the originator of any signals. Note that by the originator of a signal we only mean that the originator's spreading code is modulated in the signal, implying that, whenever it is received, the receiver would think it is from the satellite. The originator is not always the agent that actually generates the signal as attackers can also generate signals with the same C/A codes.

A GNSS receiver implements a *localisation algorithm* that takes a received signal as input and calculates a coordinate and a time point if possible. We denote the algorithm by function $loc : \mathcal{G} \to \mathcal{L} \times \mathcal{T}$. In practice, the output of a localisation algorithm is in the form of a triple consisting of a coordinate, an accuracy in meters and time. The coordinate and the accuracy define a round area centred at the coordinate with a radius of the accuracy. Since our purpose is to evaluate the integrity of GNSS signals, we assume that localisation algorithms always calculate locations with perfect accuracy. In other words, the accuracy output by a receiver is zero. For the same reason, we also omit the implementation difference between receivers. The notations mentioned are summarised in Table 3.1.

3.2.3 Signal integrity

In this section, we propose the first formal definition of the integrity of GNSS signals. Intuitively, when a received signal is free of spoofing, we say that the integrity of the signal is preserved, meaning that the signal has not been modified maliciously by the attacker. In the rest of this chapter, we also say that a signal is

integrous when it preserves signal integrity. In other words, an integrous signal is generated by a satellite and without artificial interference, e.g., replaying, before reaching the receiver.

For a received signal, the key point of verifying its integrity is to calculate the corresponding reference signal which is supposed not to be spoofed. We can make use of the following two conditions to identify the reference signal. First, the time between the generation of the reference signal and its arrival at the receiver should be equal to the amount of time required to travel the distance between its originator and the receiver by the speed of light. Second, it should suffer the correct amount of attenuation, e.g., $\eta(S, \ell, t)$, during the transition. Let $|\ell, \ell'|$ be the Euclidean distance between two positions ℓ and ℓ'. Based on the above discussion, signal integrity can be formally defined as follows:

Definition 3.1 (Signal integrity)**.** *Given a received signal $s(\ell, t)$, we say that $s(\ell, t)$ is integrous if and only if for each $sig' \in sigCom(s(\ell, t))$, there exists $t' \in \mathcal{T}$ such that*

$$(sig' = \eta(ori(sig'), \ell, t') \diamond sig(ori(sig'), t')) \wedge \ (c \cdot (t - t') = |\xi(ori(sig'), t'), \ell|)$$

where c is the speed of light.

In the rest of this chapter, we use $\mathcal{I}_{s(\ell,t)}$ to denote the proposition that "$s(\ell, t)$ *is integrous*" while $\neg\mathcal{I}_{s(\ell,t)}$ represents the negation that "$s(\ell, t)$ *is not integrous*". In practice we cannot use Definition 3.1 to verify signal integrity by computing the reference integrous signals and comparing them with the received ones. On one hand, the location of a receiver is under calculation and not available until the integrous signals have been received. Without the location, it is not possible to derive the transmission time of the received GNSS signals and thus the generation time cannot be obtained. On the other hand, the attenuation cannot be exactly quantitatively measured because of the nature of unpredictability of the ever-changing environment. Therefore, we cannot learn the set of GNSS signals that should be received.

3.2.4 Adversary model

As we mentioned before, in general the aim of an attacker is to fool a receiver to calculate a fake location. According to existing works in the literature, the attackers have two ways to achieve this purpose: *software attacks* on receivers [NLD+12] and *GNSS signal spoofing* [TPRC11].

Software attacks on receivers target at the localisation algorithms implemented on receivers. Infected by malware, the receiver can be forced to calculate incorrect coordinates. GNSS signal spoofing is to feed a receiver with simulated signals such that even the correct localisation algorithm cannot compute the right location.

In this work, we focus on the risks coming from signals, as people can protect their receivers against malware but have no control of signals. We assume that the localisation algorithm of a receiver is always well protected and free of misbehaviour. Formally, given a received signal $s(\ell, t)$ if it is integrous then we have $loc(s(\ell, t)) = (\ell, t)$.

The attackers that we consider have similar capabilities in terms of signal transmission to the attackers assumed by Tippenhauer et al. [TPRC11]. They have full control of wireless channels by blocking, intercepting, delaying and replaying GNSS signals. Furthermore, we assume that the attackers can manage to make all their signals received by the targeted receivers at any preferred time.

With regard to signal generation, we assume that the attackers can generate any GNSS signal in Θ that can be interpreted by receivers. However, the attackers cannot generate the military signals due to the encrypted $\mathsf{P(Y)}$, but it can intercept and replay them.

3.2.5 Spoofing detection methods

A spoofing detection method aims to evaluate the integrity of a given signal. It takes the measurement of a certain attribute of the signal as input and calculates a set of predicted values of the measurement. At last it decides whether the signal is integrous, by comparing the measurement to its predicted values. In the following discussion, we formally characterise spoofing detection methods and classify them.

Given a received signal $s(\ell, t)$ we denote by $Attr(s(\ell, t))$ the set of attributes of $s(\ell, t)$ that can be measured and explored by a spoofing detection method. We assume that a spoofing detection method explores only one attribute as it is designed in the literature. The value of an attribute can be measured by a receiver or calculated by other agents. For instance, the values of attributes, e.g., signal strength and Doppler shift, are calculated by receivers while others, e.g., power correlation of signals from two satellites, are not provided directly by receivers. We denote by $m_\alpha(s(\ell, t))$ the value of attribute $\alpha \in Attr(s(\ell, t))$ of $s(\ell, t)$. The domains of the measurements are different between attributes. To be generic, we use $dom(\alpha)$ to denote the domain of α. Note that for the sake of simplicity, we assume that a measurement has just a single value in its corresponding domain, while in practice the measurement of an attribute might be of different forms, e.g., a set of values in the domain. Our approach given below can be easily extended to capture these cases.

We observe that a spoofing detection method actually realises three sequential steps: generating reference measurement, validating current measurements and assessing signal integrity. In Figure 3.1, we depict these three steps as well as their output which is the input of the following steps. We address them and the meaning of their outputs one by one in the rest of this section.

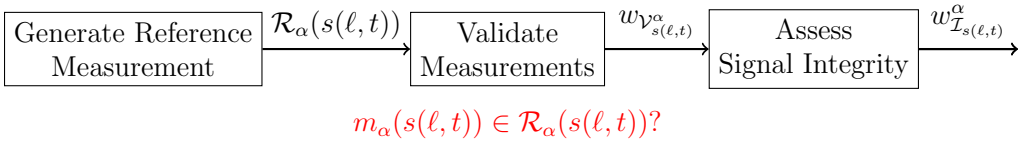


Figure 3.1: The sequential steps of a spoofing detection method.

Step 1: Generate reference measurements. Given an attribute, a spoofing detection first calculates a set of values that should contain its measurement if the

received signal is integrous (called *reference set*). Different detection methods have distinctive ways to calculate their reference sets.

We recognise two basic methods. One is to make use of a sufficiently large collection of integrous signals and calculate the set of all values that occur frequently. The other approach is to use the observation that the measurements of some attributes change over time in a fixed pattern. Based on a number of past signals the value of the current signal can thus be predicted. Based on the distinction between these two methods, we can divide spoofing detection methods into two categories: *stateless* and *stateful*. Let $\mathcal{R}_\alpha(s(\ell,t)) \subseteq dom(\alpha)$ be the calculated reference set of attribute α of signal $s(\ell,t)$. Stateless and stateful detection can be formally defined as the following.

Definition 3.2 (Stateless spoofing detection)**.** *Given a received signal $s(\ell,t)$, we say that a spoofing detection method on attribute $\alpha \in Attr(s(\ell,t))$ is stateless if*

- *$m_\alpha(s(\ell,t)) \in \mathcal{R}_\alpha(s(\ell,t))$ if $s(\ell,t)$ is integrous; and*
- *$\mathcal{R}_\alpha(s(\ell,t))$ is calculated by a function $f_\alpha : \mathcal{G} \to 2^{dom(\alpha)}$, i.e.,*

$$\mathcal{R}_\alpha(s(\ell,t)) = f_\alpha(s(\ell,t)). \tag{3.1}$$

Definition 3.3 (Stateful spoofing detection)**.** *Given a received signal $s(\ell,t)$, we say that a spoofing detection method on attribute $\alpha \in Attr(s(\ell,t))$ is stateful if*

- *for a given a set of past signals $\mathcal{H} = \{s(\ell_1,t_1),\ldots,s(\ell_n,t_n)\}$ ($\forall_{s(\ell_i,t_i)\in\mathcal{H}}\ t_i < t$), $m_\alpha(s(\ell,t)) \in \mathcal{R}_\alpha(s(\ell,t))$ if $s(\ell,t)$ is integrous and $s(\ell_i,t_i))$ is integrous for any $s(\ell_i,t_i) \in \mathcal{H}$; and*
- *$\mathcal{R}_\alpha(s(\ell,t))$ is calculated by a n-ary function $f_\alpha : \mathcal{G}^n \to 2^{dom(\alpha)}$, i.e.,*

$$\mathcal{R}_\alpha(s(\ell,t)) = f_\alpha(s(\ell_1,t_1),\ldots,s(\ell_n,t_n)). \tag{3.2}$$

In a stateless spoofing detection method a reference set is computed based on the received signal whose integrity is under evaluation. The reference set in a stateful detection method relies on some past signals. The integrity of the past signals determines the correctness of the reference set to be computed in a stateful detection method. In the definitions, we rely on the casual relation that a measurement falls in its reference set is caused by the fact that the signal is integrous. However, the related works in the literature usually take the opposite but incorrect direction, i.e., the integrity of a signal is concluded from the measurements of its attributes.

Step 2: Validate measurements. After calculating the reference set, a spoofing detection method checks whether the input measurement is in the reference set. If it is the case, we say that the measurement is *valid*. We use $\mathcal{V}^\alpha_{s(\ell,t)}$ to represent the proposition that "$m_\alpha(s(\ell,t))$ *is valid*". The notion of valid measurement is (implicitly) used by almost all existing spoofing detection methods. We formally define it in this work.

In practice, a reference set predicts a measurement considering an average intensity of natural environment interference on signal during transmission. This can lead to

incorrect validity of measurement in the cases where the interference (abnormally) deviates from the average. This means that the measurement should be valid once the interference is normal. If we can learn how much the deviation of the current interference is from the average, then there will be a way to obtain the corresponding value to the average case. However, the impact of the interference cannot be measured. Therefore, it is undesirable to have a definite conclusion that a measurement is invalid once it is out of the reference set. Instead, since subjective logic opinions can allow us to capture the uncertainty caused by the environmental interference, we express the conclusion of a detection method on the validity of $m_\alpha(s(\ell,t))$ by an opinion. It is denoted by $w_{\mathcal{V}^\alpha_{s(\ell,t)}}$ and called the *validity opinion* of $s(\ell,t)$ on attribute α.

Step 3: Assess signal integrity. At last, a spoofing detection method assesses the integrity of received signals based on the validity of the measurements.

The output of a spoofing detection method is usually qualitative in the literature, which is not correct in reality. This is mainly because: 1) unpredicted environmental interference on signals leads to uncertainty of measurement validity; 2) there does not exist a definite causal relationship from measurement validity to signal integrity. For instance, some attackers can generate signals with valid measurements if they have access to powerful simulators. In such situations measurements are valid but signals are spoofed. False negative/positive ratios are thus defined to estimate the frequency of such situations and assess the performance of the detection in the literature.

In our approach, we use a subjective logic opinion to capture the uncertainty about the integrity of a signal. Given $s(\ell,t)$, we denote the opinion on its integrity by $w^\alpha_{\mathcal{I}_{s(\ell,t)}}$ and call it an *integrity opinion*.

Summary. Based on the above discussion, upon the receipt of the measurement of an attribute α, we can summarise the three steps that a spoofing detection method sequentially performs as follows:

1. Calculate the reference set $\mathcal{R}_\alpha(s(\ell,t))$;
2. Evaluate the validity of $m_\alpha(s(\ell,t))$ according to $\mathcal{R}_\alpha(s(\ell,t)$, i.e., $w_{\mathcal{V}^\alpha_{s(\ell,t)}}$;
3. Infer the opinion on the integrity of $s(\ell,t)$ based on $w_{\mathcal{V}^\alpha_{s(\ell,t)}}$, i.e., $w^\alpha_{\mathcal{I}_{s(\ell,t)}}$.

In the literature, the calculation of reference sets in the first step has been extensively discussed. In this chapter, we take it as given. We proceed with how to obtain the validity of measurements in the second step (Section 3.3) and how to derive the integrity of signals in the third step (Section 3.4).

3.3 Deriving Validity Opinions

In this section, we give a method to calculate the validity opinion of an attribute given a received signal by taking into account the environmental interference. Essentially, we develop a function mapping $m_\alpha(s(\ell,t))$ and $\mathcal{R}_\alpha(s(\ell,t))$ to the opinion $w_{\mathcal{V}^\alpha_{s(\ell,t)}}$ for any signal $s(\ell,t)$.

Our main idea is to find an appropriate function degrading the belief on the validity of a measurement in terms of its distance to the reference set. The intuition behind this is that environmental interference with larger variation from the average is less common. A larger variance indicates that a measurement is farther away that from the reference set and thus it is less probable that the measurement is valid. There are two necessary elements in the above observation, namely, the distance of a measurement to the reference set and the degradation function.

3.3.1 Distance of measurements to reference sets

Suppose that the distance between any two elements in $dom(\alpha)$, e.g., x and x', is given as $\| x - x' \|$. The calculation and domains of the distances may vary between attributes. We assume that the distances are normalised into real numbers, i.e., $\| x - x' \| \in \mathbb{R}$. The distance of a measurement from a reference set is assigned zero if it is in the set. Otherwise, it is set as the minimum distance of the measurement to the values in the reference set. Let $d_\alpha(s(\ell,t))$ be the distance between $m_\alpha(s(\ell,t))$ and $\mathcal{R}_\alpha(s(\ell,t))$. Then it can be defined as follows:

$$d_\alpha(s(\ell,t)) = \begin{cases} 0 & m \in R \\ \min\limits_{v \in R} \| m - v \| & m \notin R \end{cases} \tag{3.3}$$

where $m = m_\alpha(s(\ell,t))$ and $R = \mathcal{R}_\alpha(s(\ell,t))$.

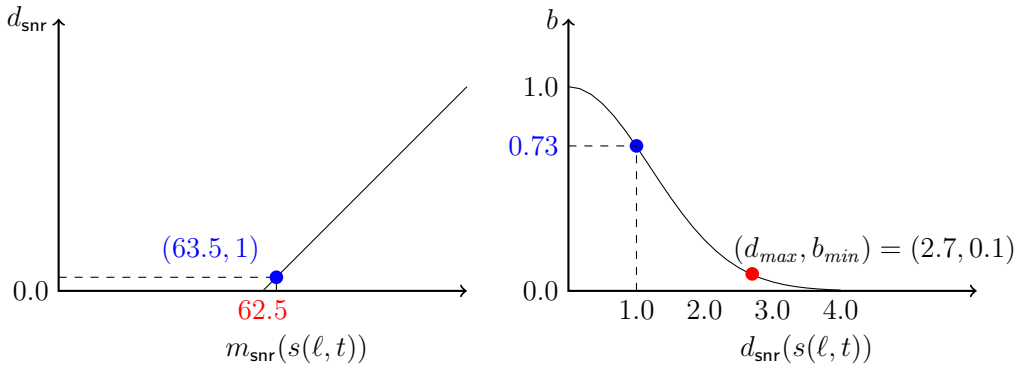
3.3.2 Degradation function

The degradation function should be smooth and be compatible with the probability distribution of the environmental interference suffered by the given signal. Note that the choice of the distribution influences the accuracy of the validity opinion and should be carefully assessed with extensive analysis, e.g., using sufficiently large number of samples. We observe that the measured values of most attributes mentioned in the literature fit normal distributions best, e.g., signal strengths and clock offsets. Although some attributes may fit different distributions, in the following we take the normal distribution as an example to define the degradation function. The main idea can be adapted to other distributions. Assume $w_{\mathcal{V}^\alpha_{s(\ell,t)}} = (b, d, u, 0.5)$. The base rate is set to 0.5 so as to express that we have no preference. The other three parameters can be computed as follows:

$$b = e^{-\frac{d_\alpha(s(\ell,t))^2}{2 \cdot var^2}}; \quad d = 1 - b; \quad u = 0 \tag{3.4}$$

where *var* represents the variance required by the original normal distribution and it determines how fast b drops along with $d_\alpha(s(\ell,t))$. The uncertainty u can be interpreted as the confidence in the existence of the normal distribution. As we have already assumed its existence, we assign 0 to uncertainty u.

We can determine the value of *var* if a distance and the corresponding belief are given. In our method, we take the maximum distance allowed for a measurement and assign the minimum belief to it. Let d_{max} be the maximum allowed distance to



(a) The function of d_{snr}. (b) The function of the belief in $w_{\mathcal{V}_s^{\mathsf{snr}}}$.

Figure 3.2: Example 3.1.

the reference set and b_{min} be the corresponding minimum belief. We can calculate *var* as follows:

$$var = \frac{d_{max}}{\sqrt{-2 \cdot \ln b_{min}}}. \tag{3.5}$$

We take the attribute – signal-to-noise ratio – as an example to explain the calculation of validity opinions.

Example 3.1. *Signal-to-noise ratios (*snr*) measure the power ration between signals and background noise. The spoofing detection method based on signal-to-noise ratios is stateless since the reference values do not depend on any past signals. According to our analysis on collected integrous GPS signals, we set the reference set* $\mathcal{R}_{\mathsf{snr}}(s(\ell,t)) = [0, 62.5]$ *for any received signal* $s(\ell,t)$.

Suppose a received signal s *and* $m_{\mathsf{snr}}(s) = 63.5$. *According to Equation 3.3, we draw the function of* $d_{\mathsf{snr}}(s)$ *in Figure 3.2(a). Then we have* $d_{\mathsf{snr}}(s) = 63.5 - 62.5 = 1$. *If we set a belief of 0.1 to the distance* 3.7, *then we have the degradation function of the shape shown in Figure 3.2(b). Thus, in terms of Equation 3.4, the validity opinion is* $(0.73, 0.27, 0, 0.5)$.

3.4 Inferring Signal Integrity

In this section, we show how to derive the integrity opinion of a signal based on the measurement validity of one of its attributes. We study the causal relationships between measurement validity and signal integrity, based on which conditional reasoning can be used. Since stateless and stateful methods have different causal relationships, they require different methods to derive integrity opinions.

3.4.1 Stateless spoofing detection

In a stateless spoofing detection method, e.g., on attribute α, a reference set is calculated in such a way that as long as a signal is integrous, its measurements must be valid (see Definition 3.2). Therefore, given a signal $s(\ell,t)$, the following

conditional relationship holds:

$$\mathcal{I}_{s(\ell,t)} \rightarrow \mathcal{V}^{\alpha}_{s(\ell,t)}. \tag{3.6}$$

The validity opinion $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}}$ has already been calculated based on the methodology given in Section 3.3. Thus the integrity opinion of $s(\ell,t)$ can be considered as the abduced opinion on the validity of the measurement.

In the abduction, we need two *a priori* conditional opinions on the measurement validity when the signal is integrous or spoofed and the *a priori* probability that the signal is integrous before its reception. Let $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\mathcal{I}_{s(\ell,t)}}$ and $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\neg\mathcal{I}_{s(\ell,t)}}$ be the opinions on the validity of the measurement when the signal is integrous or spoofed, respectively. We set the base rate $a(\mathcal{I}_{s(\ell,t)})$ to 0.5 to indicate no *a priori* knowledge about the integrity of the signal. It is a conservative choice as we want to eliminate the interference of artificial preference as much as possible. Using the abduction operator in subjective logic (i.e., $\overline{\circledcirc}$), we can calculate the opinion on the truth of $\mathcal{I}_{s(\ell,t)}$ as follows:

$$w^{\alpha}_{\mathcal{I}_{s(\ell,t)}} = w_{\mathcal{V}^{\alpha}_{s(\ell,t)}}\,\overline{\circledcirc}(w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\mathcal{I}_{s(\ell,t)}}, w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\neg\mathcal{I}_{s(\ell,t)}}, a(\mathcal{I}_{s(\ell,t)})). \tag{3.7}$$

Example 3.2. *Let us continue the Example 3.1. Suppose we have the inputs as follows:*

$$w_{\mathcal{V}^{\mathsf{snr}}_{s}|\mathcal{I}_s} = (0.94, 0.03, 0.03, 0.5); \qquad w_{\mathcal{V}^{\mathsf{snr}}_{s}|\neg\mathcal{I}_s} = (0.10, 0.80, 0.10, 0.5).$$

Then the integrity opinion of s with regard to signal-to-noise ration will be

$$w^{\mathsf{snr}}_{\mathcal{I}_s} = w^{\mathsf{snr}}_{\mathcal{V}_s}\overline{\circledcirc}(w_{\mathcal{V}^{\mathsf{snr}}_{s}|\mathcal{I}_s}), w_{\mathcal{V}^{\mathsf{snr}}_{s}|\neg\mathcal{I}_s}, a(\mathcal{I}_s)) = (0.54, 0.25, 0.21, 0.50).$$

3.4.2 Stateful spoofing detection

In a stateful spoofing detection method, e.g., on attribute α, a reference set is calculated based on a set of past signals. For the sake of simplicity, we assume that a stateful detection method only makes use of one past signal. However, our method given below can be generalised to other cases.

For a signal $s(\ell,t)$, let $s(\ell',t')$ $(t' < t)$ be the past signal based on which $\mathcal{R}_{\alpha}(s(\ell,t))$ is calculated. According to Definition 3.3, we can see that a reference set is computed in a specific way such that once past signals and the signal to be verified are both integrous, the corresponding measurement is valid. This gives rise to the following conditional relation for signal $s(\ell,t)$:

$$\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)} \rightarrow \mathcal{V}^{\alpha}_{s(\ell,t)}. \tag{3.8}$$

We cannot derive the integrity opinion $w^{\alpha}_{\mathcal{I}_{s(\ell,t)}}$ using the method given for stateless spoofing detection methods due to the involvement of the integrity of the past signals. In probability theory, if we can learn the joint probabilities $\Pr(\mathcal{I}_{s(\ell',t')}, \mathcal{I}_{s(\ell,t)})$ and $\Pr(\neg\mathcal{I}_{s(\ell',t')}, \mathcal{I}_{s(\ell,t)})$, then the probability $\Pr(\mathcal{I}_{s(\ell,t)})$ can be calculated by summing them up. This calculation is called *marginalisation*. In subjective logic if we learn the beliefs on $\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)}$ and $\neg\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)}$, then the opinion on $\mathcal{I}_{s(\ell,t)}$

can be computed in a similar way. Let I be the following multinomial frame made of $\mathcal{I}_{s(\ell',t')}$ and $\mathcal{I}_{s(\ell,t)}$:

$$I = \{\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)}, \neg\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)}, \mathcal{I}_{s(\ell',t')} \wedge \neg\mathcal{I}_{s(\ell,t)}, \neg\mathcal{I}_{s(\ell',t')} \wedge \neg\mathcal{I}_{s(\ell,t)}\}.$$

Let w_I be the multinomial opinion on I. Using the above causal relationship, we can calculate w_I based on the measurement validity through the abduction reasoning. As w_I contains the beliefs on $\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)}$ and $\neg\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)}$, we can compute the integrity opinion on $\mathcal{I}_{s(\ell,t)}$. Specifically, the calculation can be described in the following two steps:

1. Compute w_I based on $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}}$. The computation is an abductive reasoning from $\mathcal{V}^{\alpha}_{s(\ell,t)}$. Let $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|I}$ be the set of *a priori* conditional opinions on $\mathcal{V}^{\alpha}_{s(\ell,t)}$ when each proposition in I is true, i.e., $\{w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|x} \,|\, x \in I\}$. This calculation is as follows:
$$w_I = w_{\mathcal{V}^{\alpha}_{s(\ell,t)}} \overline{\circledcirc}(w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|I}, \vec{a}_I). \tag{3.9}$$

2. Compute $w^{\alpha}_{\mathcal{I}_{s(\ell,t)}}$ based on w_I. Suppose $w_I = (\vec{b}, u, \vec{a})$ and $w^{\alpha}_{\mathcal{I}_{s(\ell,t)}} = (b, d, u, a)$, then the opinion can be calculated as follows:
$$b = \vec{b}(\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)}) + \vec{b}(\neg\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)}); \tag{3.10}$$
$$u = u; \quad d = 1 - b - u; \tag{3.11}$$
$$a = \vec{a}(\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)}) + \vec{a}(\neg\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)}). \tag{3.12}$$

The base rate vector $\vec{a}$ expresses the *a priori* probability distribution on the four propositions in I. Note that $\mathcal{I}_{s(\ell',t')}$ and $\mathcal{I}_{s(\ell,t)}$ are independent as the signals $s(\ell,t)$ and $s(\ell',t')$ do not depend on each other and can be generated by two different sources. As $s(\ell',t')$ is a past signal, we assume that its integrity opinion has already been calculated, i.e., $w_{\mathcal{I}_{s(\ell',t')}}$. The expectation probability of $\mathcal{I}_{s(\ell',t')}$, i.e., $E(w_{\mathcal{I}_{s(\ell',t')}})$, is thus the *a priori* probability of $\mathcal{I}_{s(\ell',t')}$ being true. Recall that we set $a(\mathcal{I}_{s(\ell,t)})$ to 0.5 to express the absence of any knowledge about $\mathcal{I}_{s(\ell,t)}$ being true. We can calculate $\vec{a}$ as follows:

$$\vec{a}(\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)}) = E(w_{\mathcal{I}_{s(\ell',t')}}) \cdot 0.5; \tag{3.13}$$
$$\vec{a}(\mathcal{I}_{s(\ell',t')} \wedge \neg\mathcal{I}_{s(\ell,t)}) = E(w_{\mathcal{I}_{s(\ell',t')}}) \cdot 0.5; \tag{3.14}$$
$$\vec{a}(\neg\mathcal{I}_{s(\ell',t')} \wedge \mathcal{I}_{s(\ell,t)}) = (1 - E(w_{\mathcal{I}_{s(\ell',t')}}) \cdot 0.5; \tag{3.15}$$
$$\vec{a}(\neg\mathcal{I}_{s(\ell',t')} \wedge \neg\mathcal{I}_{s(\ell,t)}) = (1 - E(w_{\mathcal{I}_{s(\ell',t')}}) \cdot 0.5. \tag{3.16}$$

Some *a priori* conditional opinions are applied during the inference of signal integrity. They should be assessed properly in order to guarantee the correctness of integrity opinions. We propose an approach to determine their values in the following section.

3.4.3 Determining the conditional opinions

We can divide the conditional opinions used in Section 3.4.2 into two classes according to whether spoofed signals are involved, which are *integrous signal based*

(isb) and *spoofed signal based* (ssb). Specifically, the opinions of the form of $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\mathcal{I}_{s(\ell,t)}}$ and $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\mathcal{I}_{s(\ell',t')}\wedge\mathcal{I}_{s(\ell,t)}}$ belong to the former class while the later class includes those of the form of $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\neg\mathcal{I}_{s(\ell,t)}}$, $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\mathcal{I}_{s(\ell',t')}\wedge\neg\mathcal{I}_{s(\ell,t)}}$, $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\neg\mathcal{I}_{s(\ell',t')}\wedge\mathcal{I}_{s(\ell,t)}}$ and $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\neg\mathcal{I}_{s(\ell',t')}\wedge\neg\mathcal{I}_{s(\ell,t)}}$.

Determining isb conditional opinions. In practice, reference sets should be carefully chosen to ensure that the number of spoofed signals that have valid measurements should be small while most integrous signals have valid measurements. Reference sets do not contain all possible values that an integrous signal should have and there are situations where an integrous signal has an invalid measurement. The isb opinions express how likely these will not happen. Given the calculation of reference sets, we can estimate isb opinions by counting the frequency of valid measurements in a sufficiently large dataset of integrous signals.

We take $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\mathcal{I}_{s(\ell,t)}}$ as an example to illustrate the calculation which can be extended straightforwardly to the opinions used in stateful spoofing detection. Let SC be the collection of integrous signals and $P \subseteq SC$ be the set of samples whose measurements of α are valid. Let $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\mathcal{I}_{s(\ell,t)}}$ be (b, d, u, a). The base rate a expresses the *a priori* probability about the truth of $\mathcal{V}^{\alpha}_{s(\ell,t)}$ when the received signal is integrous. We set it to 0.5 when we have no knowledge about $\mathcal{V}^{\alpha}_{s(\ell,t)}$. Then the belief, disbelief and uncertainty can be computed by

$$b = \frac{|P|}{|SC|+2}, \quad d = \frac{|SC/P|}{|SC|+2}, \quad u = \frac{2}{|SC|+2}. \tag{3.17}$$

Determining ssb conditional opinions. The ssb opinions are related to spoofing scenarios. They express the opinions on the validity of measurements when some related signals are spoofed. They also describe the power of attackers with regard to tuning attributes when false signals are generated. The more powerful an attacker is, the more likely that the measurements of their spoofed signals remain valid.

The method of deriving isb opinions is applicable if we have samples of spoofed signals. However, as far as we know there is no publicly available dataset of spoofed signals. Instead, we propose an alternative method estimating ssb opinions based on the efforts required for the attackers to generate signals with valid measurements. Intuitively, the more efforts that are required, the less likely that the measurements of spoofed signals are valid.

There are many restrictions for the attackers to overcome in order to preserve the validity of a measurement, e.g., signal simulators, deployment environment and the availability of equipment. A spoofing attack demanding a simulator of 10,000 euros is harder than the ones which need simulators of 1,000 euros. The difficulty to meet a requirement can be divided into levels. For instance, the prices of equipment can be assigned to levels from *low* to *high*. Meanwhile, the importance of requirements also varies.

Let $Req = \{rq_1, \ldots, rq_k\}$ be the set of requirements and $W = \{w_1, \ldots, w_k\}$ be the set of corresponding importance where $\sum_{1\leq j\leq k} w_j = 1$. For $rq_i \in Req$, we assign one of the five scores $\{0.2, 0.4, 0.6, 0.8, 1\}$, i.e., $score(rq_i)$. Sometimes, we do not have expertise for every requirement. When we have no idea about the requirement, we set $score(rq_i)$ to 0. The sum of weighted assigned scores can be

interpreted as the votes against a successful spoofing attack while the unassigned scores can be seen as the neutral votes. Take $w_{\mathcal{V}^{\alpha}_{s(\ell,t)}|\neg\mathcal{I}_{s(\ell,t)}}$ for example. Let it be (b, d, u, a), then

$$b = \sum_{score(rq_i)\neq 0} score(rq_i) \cdot w_i; \quad d = 1 - b - u; \quad u = \sum_{score(rq_i)=0} w_i. \tag{3.18}$$

We set a as 0.5 to indicate the absence of any preference.

3.5 Combining Integrity Opinions

A received signal has a set of attributes that can be measured and explored by spoofing detection methods. According to Section 3.4, given a signal a detection method will calculate its integrity opinion. However, the integrity opinions can be different from each other. This is because:

- The conditional opinions used in spoofing detection methods are different. This leads to different integrity opinions even if the validity opinions are the same.
- Unpredictable environmental interference can cause an integrous signal to have incorrect validity opinions for certain attributes. This subsequently causes incorrect integrity opinions.
- Some attackers are able to tune some attributes of their generated signals so that the corresponding measurements remain valid. This fools the spoofing detection methods to output incorrect integrity opinions.

Thus, a combined integrity opinion is needed to deal with the difference. Furthermore, with more evidences taken into account, the combined opinions will be more reliable. The combination is very useful for location-based applications as they can customise their services based on signal integrity and take proper actions whenever spoofing is detected.

In this section, we propose three algorithms to combine the integrity opinions according to different user security requirements. A combination algorithm can be seen as a function taking a set of individual integrity opinions as input denoted by $\mathcal{W}_{\mathcal{I}_{s(\ell,t)}}$, and outputting a combined integrity opinion denoted by $w_{\mathcal{I}_{s(\ell,t)}}$. Before presenting the algorithms, we start with how to construct the set of integrity opinions, i.e., $\mathcal{W}_{\mathcal{I}_{s(\ell,t)}}$.

Recall that stateful spoofing detection methods make use of the integrity opinions on past signals. Assume that integrity opinions can be combined, we have two types of integrity opinions: combined opinions, e.g., $w_{\mathcal{I}_{s(\ell,t)}}$ and those given by individual stateless methods, e.g., $w^{\alpha}_{\mathcal{I}_{s(\ell,t)}}$. As a consequence, a stateful detection method can output two kinds of integrity opinions: *global* and *local*. A global integrity opinion is calculated using combined opinions on past signals, while a local integrity opinion is based on opinions given by a single stateless method. Given a signal, we thus have two sets of integrity opinions to combine – *global opinion set* and *local opinion set*, denoted by $\mathcal{W}^{glo}_{\mathcal{I}_{s(\ell,t)}}$ and $\mathcal{W}^{loc}_{\mathcal{I}_{s(\ell,t)}}$, respectively. In

this section, we use $\mathcal{W}_{\mathcal{I}_{s(\ell,t)}}$ to have a generic description for our algorithms. It can be substituted by either $\mathcal{W}^{glo}_{\mathcal{I}_{s(\ell,t)}}$ or $\mathcal{W}^{loc}_{\mathcal{I}_{s(\ell,t)}}$ in implementation.

3.5.1 The Veto algorithm

In safety-critical applications, failing to detect a spoofing attack can lead to severe consequence. In such situations, false alarms of spoofing are affordable but false claims of integrity are not. To meet this requirement, our idea is to give a spoofing alarm as long as one of the deployed spoofing detection methods gives an opinion indicating spoofing. We choose the integrity opinion with the minimum belief in the integrity of the signal as the combined opinion.

We introduce a relation to compare the belief in the integrity of a given signal expressed by two integrity opinions, i.e., $\preceq \subseteq \Omega \times \Omega$ where Ω is the set of all binomial opinions. An integrity opinion has less belief in the integrity of a signal than another if its expectation probability is smaller or it has a larger uncertainty when their expectation probabilities are equivalent. The relation $\preceq$ is formally defined as follows:

Definition 3.4 ($\preceq$)**.** *Given two binomial subjective opinions $w = (b, d, u, a)$ and $w' = (b', d', u', a')$, we say that w is not larger than w' (denoted by $w \preceq w'$) if*

$$E(w) < E(w') \vee (E(w) = E(w') \wedge u \geq u') \vee w = w'.$$

Recall that $\mathcal{W}_{\mathcal{I}_{s(\ell,t)}}$ is the set of integrity opinions output by spoofing detection methods. The calculation of the combined integrity opinion $w_{\mathcal{I}_{s(\ell,t)}}$ is straightforward (see Algorithm 3.1. Let $Veto : 2^{\Omega} \to \Omega$ be the Veto function, then we have

$$\begin{aligned} w_{\mathcal{I}_{s(\ell,t)}} = \; & Veto(\mathcal{W}_{\mathcal{I}_{s(\ell,t)}}) \text{ s.t.} \\ & (w_{\mathcal{I}_{s(\ell,t)}} \in \mathcal{W}_{\mathcal{I}_{s(\ell,t)}}) \wedge (\forall w \in \mathcal{W}_{\mathcal{I}_{s(\ell,t)}}, w_{\mathcal{I}_{s(\ell,t)}} \preceq w). \end{aligned}$$

The combined opinion is initially set to the maximum opinion with expectation probability as 1 and no uncertainty (line 4). Then it is compared with all the integrity opinions in $\mathcal{W}_{\mathcal{I}_{s(\ell,t)}}$ (line 6) and then set to any smaller opinion (line 7-9).

Note that past signals are mandatory for stateful spoofing detection methods to derive integrity opinions. When there are no sufficient opinions available for a stateful spoofing detection method, we set its integrity opinion in $\mathcal{W}_{\mathcal{I}_{s(\ell,t)}}$ to $(1, 0, 0, 0.5)$ to eliminate its impact on the combined opinion.

3.5.2 The Consensus algorithm

Recall that in a subjective logic opinion, the uncertainty mass can be interpreted as a confidence measurement on the correctness of the probability expectation. Given an integrity opinion, the smaller the uncertainty is, the more likely that its expectation probability of signal integrity is correct. Based on this understanding, the integrity opinions with less uncertainty should play a more important role in the combined opinion.

Algorithm 3.1 The Veto Algorithm

Input: $\mathcal{W}_{\mathcal{I}_{s(\ell,t)}}$
Output: $w_{\mathcal{I}_{s(\ell,t)}}$

Init: $w_{\mathcal{I}_{s(\ell,t)}} \leftarrow (1, 0, 0, 0.5)$;

for $w^{\alpha}_{\mathcal{I}_{s(\ell,t)}} \in \mathcal{W}_{\mathcal{I}_{s(\ell,t)}}$ **do**
  if $w^{\alpha}_{\mathcal{I}_{s(\ell,t)}} \preceq w_{\mathcal{I}_{s(\ell,t)}}$ **then**
    $w_{\mathcal{I}_{s(\ell,t)}} \leftarrow w^{\alpha}_{\mathcal{I}_{s(\ell,t)}}$
  end if
end for
return $w_{\mathcal{I}_{s(\ell,t)}}$

Intuitively, more evidences should lead to more reliable conclusions. This means that when more integrity opinions are combined, we should have more confidence in the correctness of the combined opinion. In other words, the combined opinion should have less uncertainty mass.

We make use of the opinion fusion operator $\oplus$ [Jøs12], which is also called the consensus operator, to combine integrity opinions. Recall that consensus is only applicable when the evidences giving rise to the opinions are independent. Since the measurement of an attribute do not affect that of another attribute, we can assume that attributes are independent of each other. Moreover, the fused opinion simply meets our expectation for the combined opinion, which can be derived straightforwardly from its definition. First, in the fused opinion, a larger proportion of the belief mass comes from the opinion with less uncertainty. Second, more opinions will lead to less uncertainty mass in the fused opinion. Let $\mathit{Consensus} : 2^{\Omega} \rightarrow \Omega$ be the corresponding function of the Consensus algorithm. Then we have

$$w_{\mathcal{I}_{s(\ell,t)}} = \mathit{Consensus}(\mathcal{W}_{\mathcal{I}_{s(\ell,t)}}) = \oplus_{w \in \mathcal{W}_{\mathcal{I}_{s(\ell,t)}}} w.$$

The algorithm making use of consensus can be shown in Algorithm 3.2.

Algorithm 3.2 The Consensus Algorithm

Input: $\mathcal{W}_{\mathcal{I}_{s(\ell,t)}}$
Output: $w_{\mathcal{I}_{s(\ell,t)}}$

Init: $w_{\mathcal{I}_{s(\ell,t)}} \leftarrow (0,\ 0,\ 1.0,\ 0.5)$;

for $w^{\alpha}_{\mathcal{I}_{s(\ell,t)}} \in \mathcal{W}_{\mathcal{I}_{s(\ell,t)}}$ **do**
  $w_{\mathcal{I}_{s(\ell,t)}} \leftarrow w^{\alpha}_{\mathcal{I}_{s(\ell,t)}} \oplus w_{\mathcal{I}_{s(\ell,t)}}$;
end for
return $w_{\mathcal{I}_{s(\ell,t)}}$

When there are no sufficient past integrity opinions for certain stateless spoofing detection methods, their integrity opinions are set to the vacuous opinion with uncertainty being 1. It is the neutral element of the consensus opinion, so it will have no impacts on the combined opinion.

3.5.3 The Combined algorithm

From their descriptions, it is clear that (1) the Veto algorithm is conservative in the sense that it can lead to more false alarms of spoofing; (2) while the Consensus algorithm can better reduce uncertainty it can lead to more false claims of integrity due to its use of the opinion fusion operator. To achieve a balance of the two situations, we combine the features of the two algorithms and develop a new algorithm. Different from the Veto algorithm, we do not always choose the integrity opinion with the smallest expectation probability to conclude a spoofed signal. Instead, we consider the opinions not only with sufficiently small expectation probabilities and but also with sufficiently small uncertainty. We call such integrity opinions *VETO* opinions.

Definition 3.5 ((σ, θ)-VETO opinions)**.** *Let $w = (b, d, u, a)$ be an integrity opinion and $\sigma \in [0, 1)$ and $\theta \in [0, 1)$ be the thresholds of the expectation probability and the uncertainty, respectively. It is said to be a VETO opinion if*

$$E(w) \leq \sigma \wedge u \leq \theta.$$

For each individual detection method, σ and θ can be set to different values. Let σ_α and θ_α be the predefined thresholds for the spoofing detection method on attribute α. When combining the opinions from a number of detection methods, if there exist multiple VETO opinions then their consensus is calculated and output as the combined opinion. Otherwise, if there is no VETO opinion, the Consensus algorithm is called to calculate the combined opinion. This new algorithm is called Combined as shown in Algorithm 3.3. The combined integrity opinion is initially

Algorithm 3.3 The Combined Algorithm

```
1: Input: W_{I_{s(ℓ,t)}}
2: Output: w_{I_{s(ℓ,t)}}
3: Init: w_{I_{s(ℓ,t)}} ← (0, 0, 1, 0.5);
4: for w^α_{I_{s(ℓ,t)}} ∈ W_{I_{s(ℓ,t)}} do
5:     if isVETO(w^α_{I_{s(ℓ,t)}}, σ_α, θ_α) then
6:         w_{I_{s(ℓ,t)}} ← w_{I_{s(ℓ,t)}} ⊕ w^α_{I_{s(ℓ,t)}};
7:     end if
8: end for
9: if w_{I_{s(ℓ,t)}} = (0, 0, 1, 0.5) then
10:    w_{I_{s(ℓ,t)}} ← Consensus(W_{I_{s(ℓ,t)}});
11: end if
```

set to the vacuous opinion. The function isVETO(w, σ, θ) returns true if w is a (σ, θ)-VETO opinion and false otherwise. We start with looking for VETO opinions in $\mathcal{W}_{\mathcal{I}_{s(\ell,t)}}$ and compute the consensus of them if there exist any (line 4-8). If there are no VETO opinions, $w_{\mathcal{I}_{s(\ell,t)}}$ will remain unchanged (line 9) as the uncertainty of a VETO opinion is always smaller than 1 (see Definition 3.5). Then we compute the consensus of all integrity opinions (line 10).

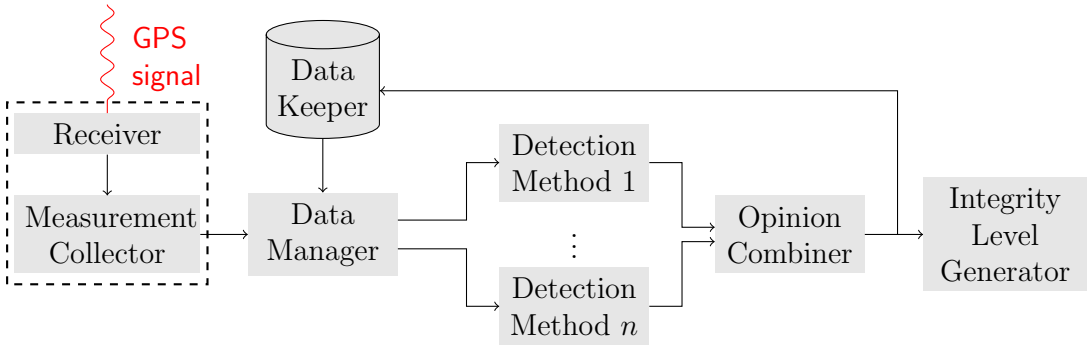


Figure 3.3: The components of the prototype.

3.6 Prototyping

We have developed a prototype based on the trust framework. It collects the measurements of received GPS (Global Positioning System) signals from receivers in real time and returns the signal integrity to users in terms of *integrity levels*.

Our prototype allows a user to customise the integrity evaluation process according to the real-time environment in order to obtain more reliable results. First, a user can disable some spoofing detection methods in certain cases when they are likely to calculate incorrect integrity opinions. For instance, when driving in a forest, a user wants to stop using detection methods relying on signal strength due to the significant fluctuations caused by trees. Second, a user can choose the algorithm to combine integrity opinions from different spoofing detection methods according to the service he is requesting. Last, a user can notify our prototype of the type of his receiver. This is necessary because receivers may differ in terms of computation power and antennas. The variants lead to different measurements of some attributes even for the same signal. In our prototype, we make a simple classification: professional and commercial-off-the-shelf, and assign different values to the *a priori* parameters used during the evaluation process.

We show in Figure 3.3 the components of our prototype. Upon receiving a signal, the receiver calculates its location. Meanwhile the *measurement collector* (MC) starts gathering the values of the attributes measured by the receiver during localisation and subsequently send them to the *data manager* (DM). We organise and record the measurements in the form of XML (Extensible Markup Language) due to its simplicity and generality. The preference of a user to customise the integrity evaluation is also added, including the spoofing detection and combination algorithms to run. The DM prepares and distributes the input for each spoofing detection method. Besides the measurements of signal attributes, other information is also included in the inputs, such as the integrity opinions of related past signals and parameters to calculate reference sets. All such information is stored and managed by the *data keeper* (DK). Integrity opinions are calculated by spoofing detection methods and then sent to the *opinion combiner* (OC) which calculates the overall integrity opinion according to the user's requirement contained in the XML file. In the end, the combined integrity opinion is transformed into an *integrity level* between 1 to 5 which is intuitive and easy for users to understand. Specifically, a signal is labelled by integrity level i if the expectation probability of

Table 3.2: The parameters used in stateless detection.

methods	reference set	d_{max}	ssb opinion
snr	$[0, 62.5]$	2.7	(0.1, 0.8, 0.1, 0.5)
dr	$[1.2829, 1.2837]$	0.004	(0.2, 0.7, 0.1, 0.5)
hd	$[0, 3.5]$	10	(0.4, 0.5, 0.1, 0.5)

the integrity opinion is between $0.2 \cdot (i-1)$ and $0.2 \cdot i$.

Note that MC should be installed on the device equipped in the receiver so as to have access to the measurements of signals (see the dashed rectangle in Figure 3.3). The other components can be deployed and run on remote agents. However, the communication between them should be well designed as users' locations are acknowledged as an important piece of private information.

3.7 Validation

In this section, we test the effectiveness of our framework based on our implemented prototype.

In our validation, we make use of four spoofing detection methods which explore the following attributes, respectively:

- Doppler ratio (dr) between the Doppler shifts of the civil signal and the military signal in a received signal.
- Signal-to-noise ratio (snr) between the power of the signal and the noise in the given RF bandwidth, which is expressed in decibels (dB).
- Height difference (hd) between the height in a calculated coordinate and the real height corresponding to the latitude and longitude in the coordinate, which is expressed in metres.
- Clock offset (cf): the time difference between the local clock of a receiver and the universal time, which is measured in seconds.

The first three spoofing detection methods are stateless while the last one is stateful as it predicts the clock offset based on one past offset and the drift speed of the local clock. In detail, suppose that the clock offset at t' is cf and the drift speed of the clock is v_{drift}. Then the predicted clock offset at time t is $preCF(t) = off + v_{drift} \cdot (t - t')$.

To learn the reference sets and related parameters, we use a dataset of 160,000 samples of integrous signals which are collected with a professional receiver JAVAD ALPHA2. Each record of the dataset stores the measurements of a signal. We choose a reference set that allows 98% of the samples to have valid measurements. Recall d_{max} is the maximal allowed distance of a measurement from the reference set. It is assigned to a value so that only 5% samples have larger distance. The corresponding minimum belief, i.e., b_{min}, is uniformly set to 0.05. Table 3.2 lists the parameters used in each stateless detection method.

The reference set of the clock offset at time t is composed of the values between $preCF(t) - 1 \times 10^{-8}$ and $preCF(t) + 1 \times 10^{-8}$. The maximum distance is set as $3 \times 10^{-8}s$ so as to ensure 5% signals with larger distance. With respect to the isb conditional opinions, as about 98% samples have valid measurements, they are set to $(0.98, 0.02, 0, 0.5)$. For the ssb conditional opinions, we assign them a preliminary opinion based on our knowledge. In our implementation, they are set to an identical opinion $(0.1, 0.8, 0.1, 0.5)$.

3.7.1 The experimental setup

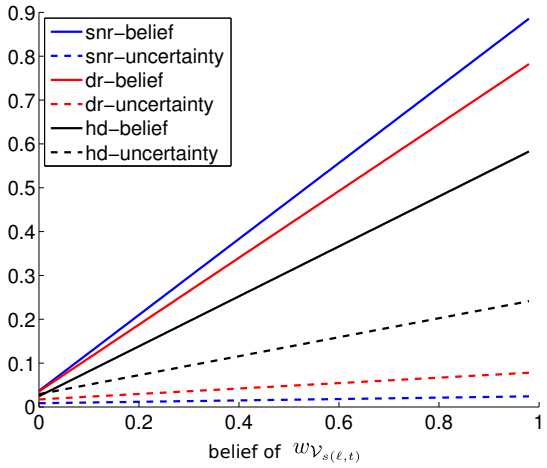
To validate our framework, we prepare three datasets of signal measurements. The first one is called *integrity dataset* storing the measurements of 25,531 integrous signals. These samples are collected using the same GPS receiver but independently from the dataset used for parameter evaluation. The second is a *spoofed dataset* and synthesised based on the integrity dataset to simulate spoofed signals. This is because no spoofed signals are publicly available. The third dataset is a *mixed dataset* with both integrous signals and spoofed signals.

The spoofed and the mixed datasets contain synthesised records for spoofed signals. The main idea to synthesise such records is to make use of the fact that the attributes of spoofed records have values deviating from those of integrous signals. Furthermore, the amount of the deviation is determined by the attackers in terms of their capabilities to tune spoofed signals. A more powerful attacker will generate signals with less deviation. We take a simple assumption that the attackers' capabilities follow the normal distribution during the construction of the spoofed dataset. To compute a record of a spoofed signal, given an item in the integrity dataset and an attribute, we first decide whether to change its value based on the corresponding *a priori* ssb conditional opinion. If yes, an extra distance is calculated following a normal distribution with d_{max} as the mean and the same variance used in the validity calculation. This extra distance is added to the distance of the original measurement and the resulted distance is used to calculate the validity opinion.

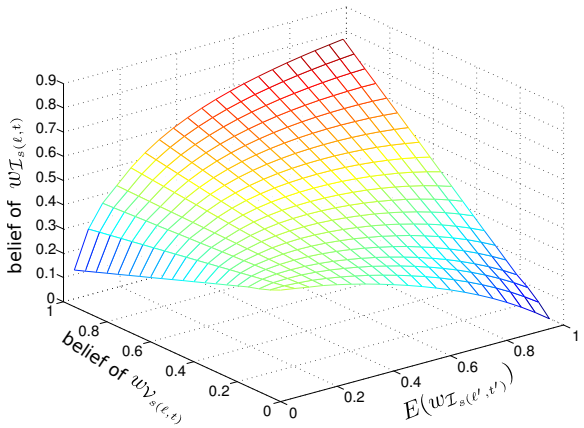
3.7.2 Experimental results

Bounds of integrity opinions. Figure 3.4 shows the change of integrity opinions with validity opinions. Figure 3.4(a) shows how the belief and uncertainty of an integrity opinion evolve with the belief of a validity opinion in the stateless methods. A general observation is that belief and uncertainty both increase linearly as the beliefs of the validity opinions grow. However, different methods have different output opinions, which are determined by their *a priori* conditional opinions. In our setting, the method snr calculates an integrity opinion with the largest belief and the smallest uncertainty than the other two spoofing detection methods.

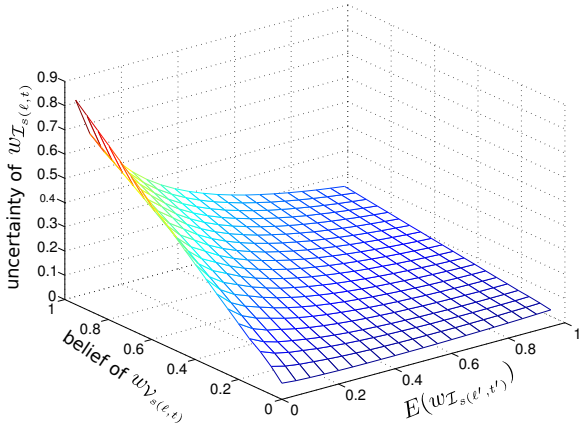
Concerning the cf stateful method, since it requires past signals, its calculated integrity opinions should change along with two parameters: the expectation probability of the past signal's integrity and the beliefs of validity opinions. Figure 3.4(b) and Figure 3.4(c) show the beliefs and uncertainty of the integrity opinions when



(a) The stateless methods.

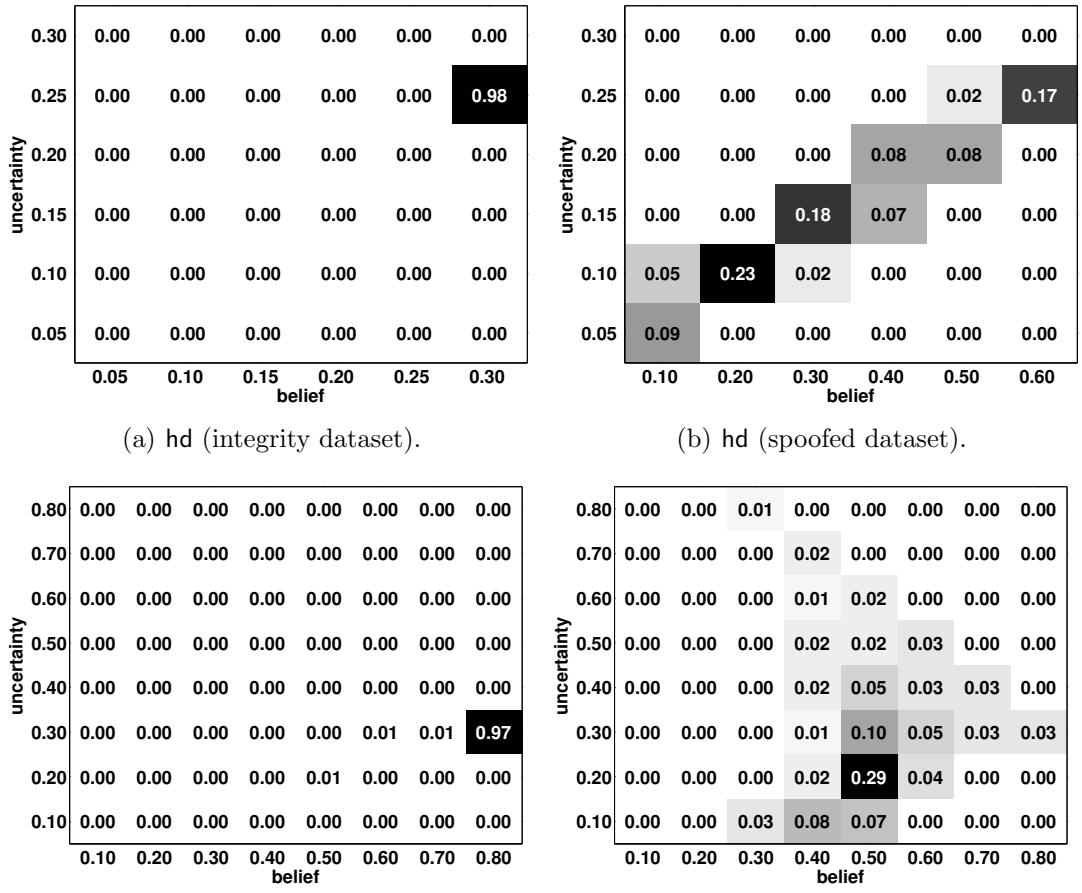


(b) The stateful method (belief).



(c) The stateful method (uncertainty).

Figure 3.4: The integrity opinions.



(a) hd (integrity dataset).

(b) hd (spoofed dataset).

(c) cf+hd (integrity dataset).

(d) cf+hd (spoofed dataset).

Figure 3.5: Integrity Opinions of individual detection methods.

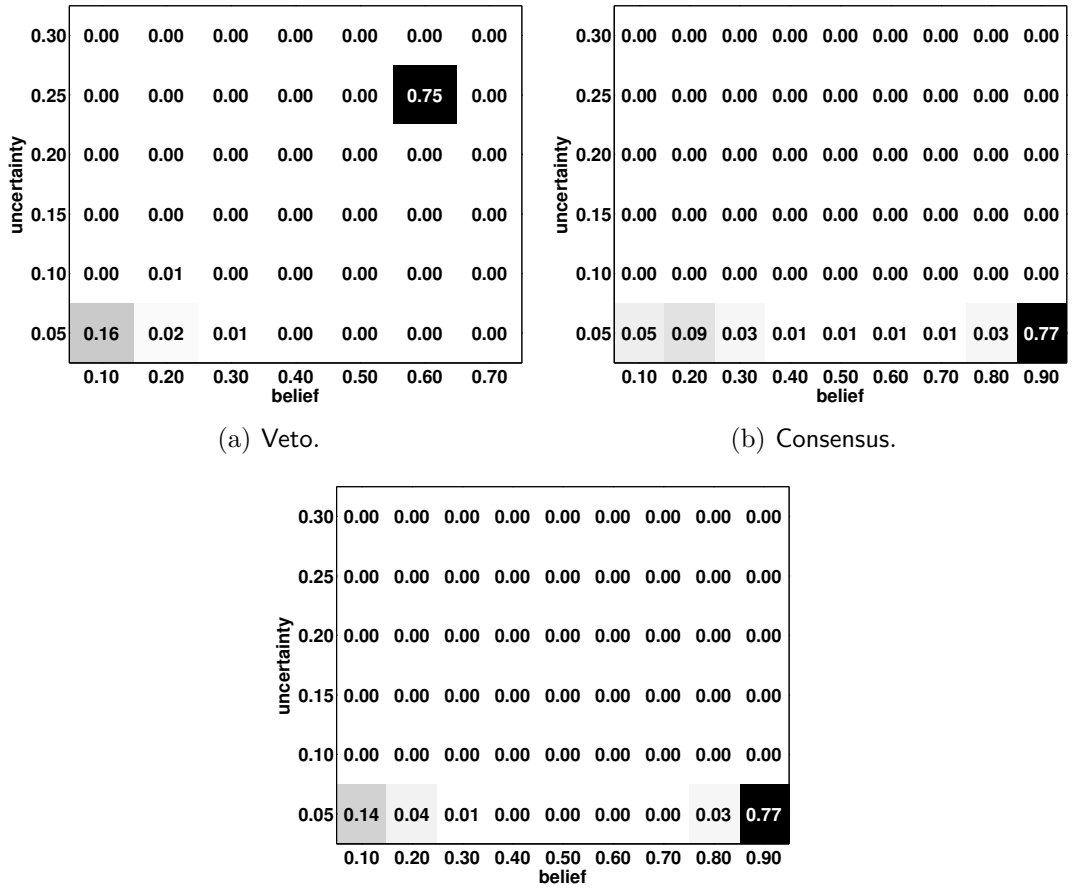
the two parameters have various values. The maximum belief value occurs when they are both 1.0 while the minimum belief is obtained when the past signal is spoofed and the current signal has a valid measurement. The maximum uncertainty is computed when the past signal is spoofed and the current measurement is valid. We use Table 3.3 to summarise the bounds of belief and uncertainty of integrity opinions for each method.

Table 3.3: Belief & uncertainty bounds of integrity opinions.

methods	**min**(b)	**max**(b)	**min**(u)	**max**(u)
snr	0.03	0.86	0.01	0.02
dr	0.09	0.78	0.02	0.07
hd	0.02	0.58	0.03	0.24
cf	0.01	0.80	0.03	0.95

Integrity opinions of spoofed and integrous signals. We study what integrity opinions spoofing detection methods calculate when signals are spoofed and integrous. To achieve this, we make use the spoofed and integrity datasets.

We divide integrity opinions into classes according to their beliefs and uncertainty. Each cell in the diagrams in Figure 3.5 corresponds to a class of opinions whose beliefs and uncertainty are bounded in certain intervals. The number labelled in



(a) Veto.

(b) Consensus.

(c) Combined.

Figure 3.6: Combined opinions of integrous signals.

each cell is the proportion of calculated integrity opinions which fall in the corresponding class. The grey level of a cell also indicates the proportion. The darker it is, the larger the proportion is. In Figure 3.5 we choose hd and cf as examples to show the distribution of integrity opinions when all signals are spoofed or integrous. Note that the cf method uses the integrity opinions of past signals given by the hd detection. We have two major observations. First, the integrity opinions of spoofed signals have much smaller beliefs and uncertainty compared to those of integrous signals. In Figure 3.5(a), we can see that 98% of the opinions given by hd on integrous signals have beliefs larger than 0.5 and uncertainty less than 0.3. However, when signals are spoofed, the beliefs of about 50% integrity opinions drops below 0.2 and the uncertainty becomes smaller than 0.15 (see Figure 3.5(b)). The opinions computed by the cf detection follow a similar pattern. Second, different methods give different opinions even for the same signals. The opinions on both datasets given by the two methods rarely overlap.

Integrity opinion combination. We use the mixed dataset to validate the performance of the combination algorithms. Intuitively, a combined algorithm is effective if it can calculate large beliefs for integrous signals and small beliefs for spoofed signals. In the mixed dataset, we have 4,748 spoofed signals out of total 25,531 samples (about 18.6%). Figure 3.6 shows the results of our three algorithms. They all successfully distinguish spoofed signals from integrous ones (with certain

Table 3.4: The average integrity opinions of integrous and spoofed signals.

	Avg. Op. (spoofed)	**Avg. Op. (Integrous)**
snr	0.23, 0.76, 0.01	0.88, 0.10, 0.02
dr	0.29, 0.67, 0.04	0.78, 0.15, 0.08
ht	0.30, 0.58, 0.12	0.58, 0.18, 0.24
cf+snr	0.38, 0.30, 0.32	0.78, 0.01, 0.20
cf+dr	0.41, 0.31, 0.28	0.77, 0.01, 0.22
cf+hd	0.46, 0.31, 0.23	0.75, 0.01, 0.24
Veto	0.08, 0.89, 0.03	0.58, 0.19, 0.23
Consensus	0.22, 0.77, 0.01	0.88, 0.11, 0.01
Combined	0.11, 0.87, 0.12	0.86, 0.12, 0.02

errors). The Veto algorithm assigns smaller beliefs and larger uncertainty to both spoofed and integrous signals, as it is rather conservative when compared with the other two methods. The Consensus algorithm assigns 77% of the signals with beliefs larger than 0.9, and assigns 14% of the signals with beliefs less than 0.2 meaning that about 4.6% of the spoofed signals are not detected. The Consensus algorithm gives uncertainty less than 0.05 to almost all the signals. It is interesting to see that the Combined algorithm gives more balanced results. When signals are integrous, a belief of 0.9 is mostly assigned which is the same as the Consensus algorithm. Meantime, for spoofed signals, it assigns a belief of 0.1 to most of them, which is comparable to the Veto algorithm and much better than the Consensus algorithm. The observations follow the design principles of the algorithms. In practice, the choice of a combination algorithm depends on applications.

In Table 3.4, we lists the average opinions computed by the individual detection methods and the combined opinions calculated by our combination algorithms. Not that we only give the belief, disbelief and uncertainty values of the opinions as the base rate is always 0.5 in our setting. We can see that the opinions given by our combination algorithms have smaller belief values compared with the individual methods. This means that the combined opinions are more reliable when signals are not integrous. Furthermore, the Veto algorithm computes the opinions with smallest belief for spoofed signals while the consensus algorithm gives the largest belief. By combining the ideas of the previous two algorithms, the Combined algorithm can enforce a balance between the two.

3.8 A Demonstrator: Location Assurance Provider

To demonstrate our trust framework for evaluating GNSS signal integrity, we present an implementation of a public service: *location assurance certification* based on our prototype in practice. *Location-based services* (LBS) are services customised according to users' locations. Delivering a service calculated with a wrong location will lead to security concerns such as privacy leakage. Take location-based local friend search as an example. Users send requests to LBS providers for the list of friends who are close to them in order to have a common activity. By feeding a user's device with false locations, attackers can learn the locations of any friends of

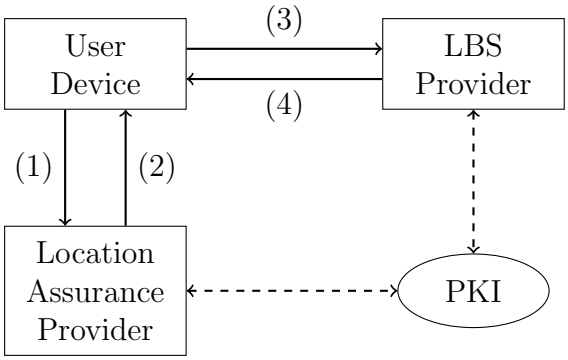
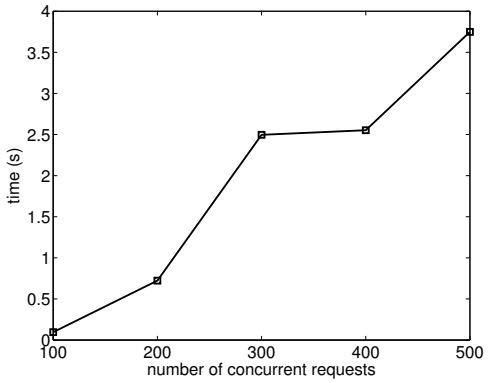


Figure 3.7: Location Assurance Provider (LAP).

the user which should not be revealed according to the user's real location. To fight against such attacks, only to protect users' device from malware is not sufficient because spoofing is still possible.

We have implemented a trusted central server called *location assurance provider* (LAP) based on our prototype to evaluate signals' integrity and issue a certificate on their integrity levels (called *location assurance certificate*). The certificate is then sent to the LBS provider who will verify and adjust its policy (e.g., stop or continue) to deliver the service according to the integrity level. Figure 3.7 shows the main steps for a user to request an LBS using location assurance certification. Before sending a request to the LBS provider, the user device first collects the measurements of received signals and contacts the LAP to evaluate their integrity (step (1)). Upon receiving the location assurance certificate (step (2)), the user sends an LBS request together with the certificate (step (3)) to the LBS provider. The provider checks the validity of the certificate and returns the service catered in terms of the integrity level attached (step (4)). To accomplish the scheme, a public key infrastructure (PKI) is required to manage the LAP's public key. Besides the LAP we also implement an Android application which runs on users' mobile devices. In fact, the application works as the measurement collector (MC in Figure 3.3) and takes charge of communicating with the LAP. We use a 3G telecommunication network to establish the connection with the LAP. According to our test, the average transition time of a message is about 2 seconds.

We test the efficiency and effectiveness of the LAP in terms of computation time and numbers of false conclusions. The LAP is run on a virtual machine with 4G RAM and an Intel Xeon E5-2640 processor. Figure 3.8(a) shows the average computation time for a request when different number of users send requests concurrently – it increases when the number of requests gets large. This is because for a request, the LAP needs two operations on the database (read parameters and store integrity opinions) which takes about 90% of the computation time. However, even with the current setting, for 500 requests, we need less than 4 seconds which is still acceptable. More efficient database techniques can improve the parallelism of the computation. Figure 3.8(b) shows the distributions of integrity levels of four spoofing detection methods and our three integration algorithms on the mixed dataset with about 18% of spoofed signals.



(a) Computation time of a request.

	1	2	3	4	5
SN	0.14	0.02	0.01	0.03	0.79
DR	0.13	0.03	0.02	0.03	0.80
HC	0.06	0.08	0.05	0.81	0.00
CB	0.00	0.00	0.18	0.04	0.78
Veto	0.19	0.02	0.02	0.78	0.00
Consensus	0.14	0.04	0.02	0.03	0.77
Combined	0.18	0.02	0.00	0.03	0.77

Integrity Level

(b) Integrity opinions.

Figure 3.8: Testing result of the LAP.

3.9 Related Work

In this section, we briefly present the state-of-the-art about spoofing countermeasures. In general they can be divided into two categories: *detection-oriented* and *defence-oriented*.

We start with the first category which is also our focus in this thesis. Some low-cost methods are proposed to detect unsophisticated spoofing [WJ03, WHD+05, MHL09, PJ08, JJBNL12]. For instance, Papadimitratos et al. [PJ08] summarise three spoofing detection tests: location inertial test, clock offset test and Doppler shift test. Inertial sensors, such as speedometers and altimeters, can be used to predict future locations based on past ones, which are usually close to the real locations. The clock offset test measures the time offset of a receiver's local clock to the system time. As clocks usually drift with a fixed ratio, future clock offsets can be computed and the real offsets should be around them. Doppler shifts are also predictable if the relative velocities of a receiver to the satellites are available.

There are also some methods that make use of more advanced attributes of GNSS signals. For example, Nielsen et al. [NBL10] monitor the correlation between the strengths of two signals from different satellites because the strengths always change independently. Psiaki et al. [POB+13] utilise the correlation between the encrypted military signals received by different receivers as the military signals transmitted by the same satellite should be physically the same even if they cannot be decrypted by civil receivers.

The above detection methods are designed under the same principle. Namely, given a signal, a method takes the measurement of a certain attribute of the signal as input, calculates the predicted values and claims the absence of spoofing when the measurement is sufficiently close to the prediction. To the best of our knowledge, the existing detection methods in the literature all belong to this category.

With regard to defence-oriented methods, their main idea is to explore cryptography. A straightforward solution in this category is to add the digital signatures of the navigation message [PCDC10, WRH12, Hum13]. Such methods all require to modify the current format of GNSS signals which as we mentioned will cost a lot of efforts and take a long period to deploy.

3.10 Conclusion

The civil signals of GNSS systems suffer from attacks such as spoofing by its design. Due to people's huge reliance on them, it is in great need to provide a method for a user at least to learn whether the received signals are trustworthy or not even if at the moment no effective defence is available. Such an evaluation should be able to handle the influences from complexed environment in practice on GNSS signals. Many spoofing detection methods have been proposed but they do not follow a correct reasoning from observation to conclusion. Moreover, due to the lack of a formal definition of signal integrity, the targets of existing detection methods are not compatible sometimes. This prevents us from exploring existing works in an effective way.

We proposed a trust framework to evaluate the integrity of GNSS signals. We identified a few problems with existing spoofing detection methods in the literature and addressed them within our framework. First, we clarified the concept of signal integrity and gave a formal definition, which is the first attempt to the best of our knowledge. Second, we precisely characterised spoofing detection methods and extracted the causal relations between measurement validity and signal integrity. We then proposed an approach to derive signal integrity while capturing its uncertainty in a natural way. Our third contribution is that we presented three ways to combine opinions from various detection methods. Last but not least, we implemented a prototype of our framework and based on which we validated our method through experiments. To demonstrate the potential value in practice, we applied our prototype to implement a public service which provides users with location assurance certificates on their received signals.

Part II

Query Privacy

4

Protecting Query Privacy

Privacy should be protected, which is nowadays a common sense. However, this request for protection is hard to satisfy, as privacy is often endangered in subtle ways. In this chapter we study how privacy leaks from queries that a user sends out, and how an adversary who knows contextual information, a form of indirect knowledge, threatens the privacy of users, which is supposed to be protected. We show by a formal framework that by exploring contextual information, attackers can derive the probability that an anonymous query belongs to a specific user. We make use of user profiles and query dependency as examples to illustrate the implementation of such attacks. By proposing a series of new metrics, we illustrate that users' query privacy can be precisely and flexibly measured. By proposing new algorithms, we show that users' query privacy can be protected according to users' privacy requirement against malicious inference from contextual information.

4.1 Introduction

The basic idea to protect users' query privacy in LBSs is to break the link between user identities and requests [BMW+09]. The straightforward protection is to remove or replace identities with pseudonyms, e.g., mix-zones [FSH09] and their variants [BLPW08]. However, for LBSs, this protection mechanism has been proved insufficient in a number of cases. Locations contained in requests can still reveal issuers' identities, since attackers can acquire users' locations through a number of methods, e.g., triangulating mobile phones' signals and localising users' access points to the Internet. In such cases, users' spatial and temporal information serves as *quasi-identifiers*. Anonymisation techniques from other research areas such as sensitive data release [GG03] are thus introduced, including k-anonymity and its different extensions (e.g., ℓ-diversity and t-closeness [LLV07, MKGV07]). Locations or time are replaced with regions or periods so that a certain number of users (at least k for k-anonymity) share the same quasi-identifier with the real issuer. The calculation of the regions or periods is termed as *generalisation* or *cloaking*. Since in practice LBS providers are usually required to offer immediate responses, we will not consider temporal generalisation in this thesis. We call a request *generalised* if the location is generalised and the user identity is removed.

In spite of the protection of various generalisation algorithms, when the adversary has access to additional information, new privacy risks will emerge. For instance, some existing generalisation algorithms are found to suffer a type of attacks called "outlier" attacks when their implementation is made public [MBFW07]. Some users can be eliminated by the adversary from the set of potential issuers of the

generalised request. This is because the algorithms cannot output the same generalised request for these users if they issued the same query at the same time as the real issuer. Information such as the implementation of generalisation algorithms is classified as *contextual information* in the literature and privacy in LBSs related to contextual information is named as *context-aware privacy* [RPB09]. Many types of contextual information have been studied so far. For example, Shin et al. [SAV08, SAV11] study user profiles and propose metrics based on k-anonymity by restricting levels of similarity among users in terms of their profiles. Mascetti et al. [CZBP06] propose the concept of *historical k-anonymity* against attacks where the adversary learns a trace of associated requests, e.g., issued by the same user.

The research on context-aware privacy usually follows a two-step approach. It starts with identifying a type of contextual information and demonstrating its impact on users' privacy and then it proceeds with developing specific privacy protection mechanisms. There are a few problems with this line of research.

- The privacy concern related to contextual information is usually illustrated in a possibilistic way with a focus on whether a type of contextual information has impact on query privacy or not. It is not clear how much impact has been exactly made by this piece of contextual information.
- Different types of contextual information are studied independently. As a result, the privacy protection mechanisms proposed are only effective for certain contextual information but not for others. In particular, with the development of information and communication techniques, new contextual information will always be identified. Such contextual information may expose users to new query privacy risks.

In this thesis, we propose a uniform framework to assess the impact of contextual information on users' query privacy. Moreover, this framework allows us to define generic privacy metrics for users to express their privacy requirements. In order to get users' query privacy protected according to their requirements, we develop new algorithms.

4.2 Our Framework

In this section, we present our new framework for query privacy analysis in LBSs. This framework allows us to precisely specify relevant components and attacks on query privacy with various contextual information.

4.2.1 Mobile users

We assume a set of users who subscribe to an LBS and use the service frequently during their movements. Let $\mathcal{U}$ be the set of such users. We use $\mathcal{L}$ to denote the set of all possible positions where a user can issue a request. The accuracy of any position $\ell \in \mathcal{L}$ is determined by the positioning devices used. We represent time as a totally ordered discrete set $\mathcal{T}$, whose granularity, e.g., minutes or seconds, is decided by the LBS provider. The function $\mathit{whereis} : \mathcal{U} \times \mathcal{T} \to \mathcal{L}$ gives the

exact position of a user at a given time. Thus, for any time $t \in \mathcal{T}$, users' *spatial distribution* is $dis_t = \{\langle u, whereis(u,t)\rangle \,|\, u \in \mathcal{U}\}$. Suppose the set of queries (e.g., the nearest gas station) supported by LBS providers is represented by $\mathcal{Q}$. An LBS request is then in the form of $\langle u, \ell, t, q\rangle \in \mathcal{U} \times \mathcal{L} \times \mathcal{T} \times \mathcal{Q}$, where $\ell = whereris(u,t)$.

4.2.2 Request generalisation algorithms

The generalisation of LBS requests is usually implemented in two ways: *centralised* and *distributed*. A centralised structure (depicted in Figure 4.1) relies on a trusted agent, the *anonymiser*, to collect users' requests and anonymise them before sending them to LBS providers. However, in a distributed implementation users cooperate with each other to construct a generalised region [GKS07, SHSH11]. The centralised framework is easy to implement and well-studied in the literature while the distributed framework requires more communications between collaborators and security analysis, e.g., with respect to *insiders*, is not well studied. For this reason, in this thesis, we make use of the centralised implementation to protect users' query privacy. In the centralised framework, normally it is assumed that the communication channels between users and the anonymiser are secure while the ones between the anonymiser and the LBS provider are public.

Figure 4.1: A centralised framework of LBSs.

Given a request $\langle u, \ell, t, q\rangle$, the anonymising server (*anonymiser*) will remove the issuer's identity (i.e., u) and replace his location (i.e., ℓ) with an area to protect his query privacy. We only consider *spatial generalisation* in this chapter as in LBSs users require instant responses. Let $2^{\mathcal{L}}$ be the power set of $\mathcal{L}$. Then the set of all possible generalised regions can be denoted by $\mathcal{R} \subseteq 2^{\mathcal{L}}$. Given $\langle u, \ell, t, q\rangle$, the anonymising server outputs a *generalised request* in the form of $\langle r, t, q\rangle$, where $r \in \mathcal{R}$ is the generalised area and $\ell \in r$. The generalisation algorithm of the anonymiser can thus be represented as a function $f : \mathcal{U} \times \mathcal{L} \times T \times \mathcal{Q} \to \mathcal{R} \times \mathcal{T} \times \mathcal{Q}$. We use the function *query* to obtain the query of a (generalised) request (i.e., $query(\langle u, \ell, t, q\rangle) = q$ and $query(\langle r, t, q\rangle) = q$).

The generalisation algorithm also takes users' privacy requirements as part of its input. In our framework, a privacy requirement is represented by a pair: a chosen privacy metric and the corresponding value (e.g., $\langle \mathsf{k-anonymity}, 5\rangle$). We use $req(\langle u, \ell, t, q\rangle)$ to represent u's privacy requirement on request $\langle u, \ell, t, q\rangle$.

4.2.3 The adversary

Privacy risks and countermeasures should be categorised according to the adversary's model and objectives [BMW+09]. For query privacy, the adversary's goal is obviously to associate issuers to their queries while the model should be defined in terms of his *knowledge* and *attack(s)* [SAV11].

The knowledge of an adversary can be interpreted as the contextual information that he has access to. We denote by $\mathcal{C}_t$ his collection of contextual information at time t. To model the knowledge that is commonly assumed accessible to the adversary, some contextual information is inherently contained in $\mathcal{C}_t$. Specifically, in this chapter, we assume that the adversary has access to the following contextual information:

i) the deployed request generalisation algorithm (i.e., f);

ii) users' spatial distributions before the current time t, i.e., $\mathcal{D}_t = (dis_{t_1}, \ldots, dis_{t_n})$ where $t_n = t$ and $\forall_{1 \leq i < n} t_i < t_{i+1}$;

iii) the probability of a user u to issue a request at a given time t is uniformly distributed, i.e., $\Pr(u|\mathcal{C}_t) = \Pr(u'|\mathcal{C}_t)$ $(\forall u' \in \mathcal{U})$.

The assumptions i) and ii) are commonly made in the literature and the assumption iii) is made according to the principle of maximum entropy [Jay57a, Jay57b]. These assumptions make a strong adversary which allows us to analyse query privacy in the worst case. The availability of dis_t enables the adversary to obtain the set of users located in any region r at time t, which is denoted as $u\ell(r, t)$.

Given a generalised request $\langle r, t, q\rangle$ and contextual information $\mathcal{C}_t$, the objective of an attack performed by the adversary on query privacy is to learn the request's issuer. In most of the cases, the adversary is not sure of the issuer. Uncertainty is thus inevitable. We use a probability distribution over users to capture his certainty and quantify the expected correctness of his attack. Let variable U be the issuer of $\langle r, t, q\rangle$. For any user $u \in \mathcal{U}$, his probability to issue the request $\langle r, t, q\rangle$ from the view of the adversary with $\mathcal{C}_t$ can be represented as $\Pr(U = u \mid \langle r, t, q\rangle, \mathcal{C}_t)$. In the following, we give one method the adversary can adopt to calculate the distribution. Through the Bayesian theorem, we have equation:

$$\begin{aligned} \Pr(U = u \,|\, \langle r, t, q\rangle, \mathcal{C}_t) &= \frac{\Pr(\langle r, t, q\rangle \,|\, u, \mathcal{C}_t)}{\Pr(\langle r, t, q\rangle, \mathcal{C}_t)} \\ &= \frac{\Pr(\langle r, t, q\rangle \,|\, u, \mathcal{C}_t) \cdot \Pr(u \,|\, \mathcal{C}_t) \cdot \Pr(\mathcal{C}_t)}{\sum_{u'} \Pr(\langle r, t, q\rangle \,|\, u', \mathcal{C}_t) \cdot \Pr(u' \,|\, \mathcal{C}_t) \cdot \Pr(\mathcal{C}_t)}. \end{aligned} \tag{4.1}$$

In the above equation, there are three new distributions. The distribution $\Pr(\mathcal{C}_t)$ measures the probability of the adversary having learned the collection of contextual information $\mathcal{C}_t$. It is difficult to evaluate its value. However, since it appears in both the numerator and the denominator, we can eliminate it from the formula. Recall that the distribution $\Pr(u \,|\, \mathcal{C}_t)$ represents the probability for user u to issue a request at time t based on the contextual information $\mathcal{C}_t$ and it is assumed as uniform. Thus, the target posterior distribution can be further simplified as:

$$\Pr(U = u \,|\, \langle r, t, q\rangle, \mathcal{C}_t) = \frac{\Pr(\langle r, t, q\rangle \,|\, u, \mathcal{C}_t)}{\sum_{u' \in \mathcal{U}} \Pr(\langle r, t, q\rangle \,|\, u', \mathcal{C}_t)}. \tag{4.2}$$

The probability $\Pr(\langle r, t, q\rangle \,|\, u, \mathcal{C}_t)$ indicates the likelihood that 'if user u generates a request at time t then the request will be generalised as $\langle r, t, q\rangle$'. This is actually a joint probability of the following two probabilities. The first is the probability that

Table 4.1: Notations.

$\mathcal{U}$	set of users
$\mathcal{T}$	set of time instances
$\mathcal{L}$	set of locations
$\mathcal{R}$	set of possible generalised regions
$q \in \mathcal{Q}$	a query supported by the LBS
$\langle u, \ell, t, q\rangle$	a request issued by u at position ℓ at time t
$\langle r, t, q\rangle$	a generalised request
$whereis(u, t)$	position of user u at time t
$f(\langle u, \ell, t, q\rangle)$	an algorithm computing generalised queries
dis_t	spatial distribution of users in $\mathcal{U}$ at time t
$u\ell(r, t)$	set of users located in region r at time t
$req(\langle u, \ell, t, q\rangle)$	user u's privacy requirement on $\langle u, \ell, t, q\rangle$
$query(\langle r, t, q\rangle)$	the query of $\langle r, t, q\rangle$

user u issues the request $\langle u, whereis(u, t), t, q\rangle$ when he sends a request at t. It can also be formulated as the probability that u chooses query q at time t to consult the LBS provider. We call this probability the *a priori probability* of user u. The second probability is the likelihood that the area generalisation algorithm outputs a region r for $whereis(u, t)$. We use $\Pr_u(q \,|\, \mathcal{C}_t)$ and $\Pr(f(\langle u, whereis(u, t), t, q\rangle) = \langle r, t, q\rangle)$ to represent these two probabilities, respectively. Based on the above discussion, formally we have

$$\Pr(\langle r, t, q\rangle \,|\, u, \mathcal{C}_t) = \Pr_u(q \,|\, \mathcal{C}_t) \cdot \Pr(f(\langle u, whereis(u, t), t, q\rangle) = \langle r, t, q\rangle). \tag{4.3}$$

We assume that the generalisation algorithms mentioned in this chapter are *deterministic*. In other words, there is always a unique generalised request corresponding to each LBS request, which leads to $\Pr(f(\langle u, whereis(u, t), t, q\rangle) = \langle r, t, q\rangle)$ being either 1 or 0. Furthermore, given an LBS request and a generalised request, the value of this probability is available to the adversary as generalisation algorithms are public. Therefore, the key of query privacy analysis is to calculate $\Pr_u(q \,|\, \mathcal{C}_t)$ for any query $q \in \mathcal{Q}$.

The calculation of $\Pr_u(q \,|\, \mathcal{C}_t)$ depends on $\mathcal{C}_t$, i.e., the available contextual information. In the following discussion, we give the instantiations of our framework when two different types of contextual information are added into the adversary's knowledge, i.e., user profiles (see Section 4.3) and query dependency (see Section 4.4). In this way, we not only show that our framework can handle the contextual information that has been studied (i.e., user profiles), but also demonstrate that it is generic to cope with new context (i.e., query dependency). The important notations are summarised in Table 4.1.

4.2.4 Classifying contextual information

From the above discussion, we can see that the adversary can learn new knowledge along with time and contextual information will evolve along with time. For instance, the contextual information about users' spatial distributions (i.e., $\mathcal{D}_t$) records the sequence of the snapshots of mobile users' locations up to time t and

this knowledge keeps growing with time. However, we also notice that certain contextual information remains stable over time such as user mobility patterns and user profiles.

According to this observation, we classify contextual information into two classes: *static* and *dynamic*. Formally they can be defined as follows:

Definition 4.1 (Static & dynamic context)**.** *Let $\varphi_t \in \mathcal{C}_t$ be the value of a type of contextual information at time t. We say that the contextual information is static if and only if for any two time points t and t' in $\mathcal{T}$, $\varphi_t = \varphi_{t'}$. Otherwise, the contextual information is dynamic.*

Note that in practice the above definition can be relaxed. When a type of contextual information keeps stable for a sufficiently long period, we can also consider it as static. For instance, a user profile can be interpreted as static even though the user may change his job as switching jobs is not frequent.

In Figure 4.2, we classify the contextual information mentioned in this chapter. To attack query privacy, the adversary usually combines different contextual information. For instance, when associated requests are explored [CZBP06, BMW+09, DRRW10a], request generalisation algorithms and users' real-time spatial distribution are also part of the adversary's knowledge.

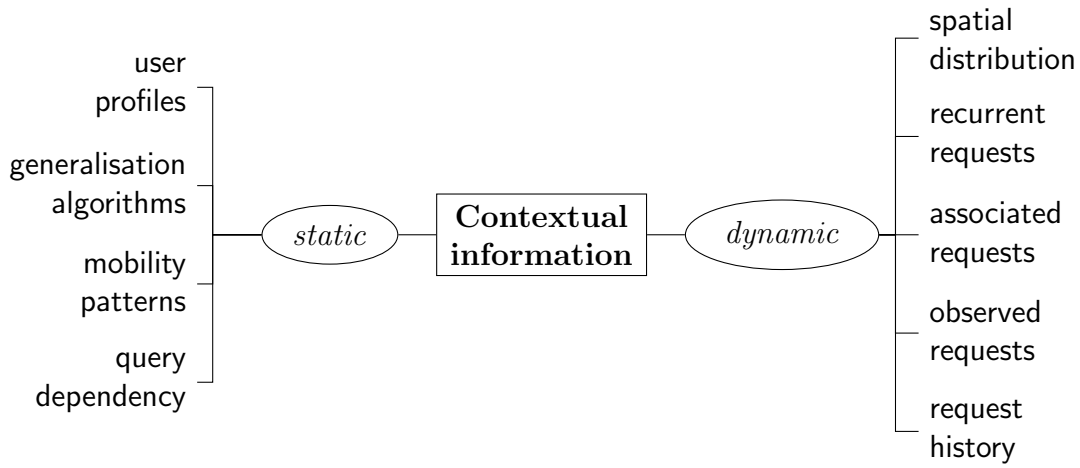


Figure 4.2: A classification of contextual information.

4.3 Privacy Analysis based on User Profiles

In this section, we demonstrate the implementation of our framework when user profiles are explored by the adversary. Although user profiles and their impact on query privacy have been discussed by Shin et al. [SAV08], they do not describe precisely how to exploit user profiles and quantify the amount of benefits gained by the adversary. On the contrary, with our framework we can formally define the attack and use a posterior probability distribution to describe the adversary's knowledge about the issuer.

As discussed in Section 4.2, given a generalised request $\langle r, t, q\rangle$ the key of query privacy analysis is to compute users' *a priori* probabilities, e.g., $\Pr_u(q|\mathcal{C}_t)$. Before presenting the calculation, we start with formulating the adversary' knowledge. User profiles are associated with a set of attributes, e.g., contact attributes (zip codes, addresses), descriptive attributes (age, nationalities, jobs) and preference attributes (hobbies, moving patterns) [SAV08]. The values of these attributes can be categorical (e.g., nationality) or numerical (e.g., salary, age). Let $\langle a_1 : \mathcal{A}_1, \ldots, a_m : \mathcal{A}_m\rangle$ be the list of the attributes where a_i is the name of the attribute and $\mathcal{A}_i$ is its domain. The profile of user u can be represented as a tuple of values each of which corresponds to an attribute. Let $\Phi_u = \langle \alpha_1, \ldots, \alpha_m\rangle \in \mathcal{A}_1 \times \ldots \times \mathcal{A}_m$ be the tuple where α_i is the value of a_i and denoted by $\Phi_u^{a_i}$. Thus the contextual information learnt by the adversary at time t can be represented as the following:

$$\mathcal{C}_t = \{\mathcal{D}_t, f, \{\Phi_u \,|\, u \in \mathcal{U}\}\}. \tag{4.4}$$

Our main idea to calculate $\Pr_u(q|\mathcal{C}_t)$ is to compute the relevance of user u's profile to each query and compare the relevance to q with those to other queries. Given an attribute a_i, we can discretise its domain $\mathcal{A}_i$ into intervals if it is numerical or divide the domain into sub-categories if it is discrete. For instance, the domain of attribute address can be categorised in terms of districts while the numerical values of salary can be discretised into three intervals, such as '≤ 1000', '$1000-5000$' and '≥ 5000'. Note that the intervals are mutually exclusive and their union is equal to the original domain.

With the list of the intervals, we can transform the value of an attribute into a vector of binary values based on which interval or category it belongs to. Suppose $\mathcal{A}_i$ is divided into the list of intervals $(\mathcal{A}_i^1, \ldots, \mathcal{A}_i^k)$ where for any $1 \leq x, y \leq k$, $\mathcal{A}_i^x \cap \mathcal{A}_i^y$ and $\cup_{1\leq j\leq k}\mathcal{A}_i^j = \mathcal{A}_i$. Let $\vec{\Phi}_u^{a_i}$ be the vector of $\Phi_u^{a_i}$ and $[\vec{\Phi}_u^{a_i}]_j$ be the jth value. Thus, we have

$$[\vec{\Phi}_u^{a_i}]_j = \begin{cases} 1 & \text{if } \Phi_u^{a_i} \in \mathcal{A}_i^j; \\ 0 & \text{if } \Phi_u^{a_i} \notin \mathcal{A}_i^j. \end{cases} \tag{4.5}$$

If a user has a salary of 3000 euros, then $\vec{\Phi}_u^{\mathsf{salary}} = [0\ 1\ 0]$.

Each query $q \in \mathcal{Q}$ has a set of related attributes that determines whether it is likely for a user to issue the query q. Furthermore, for a given related attribute, its value decides the amount of likelihood. For instance, for the query asking for expensive hotels, the associated attributes should include salary, jobs and age while gender is irrelevant. Among them, a salary is much more relevant than age and moreover, a salary of more than 5000 euros is much more important than one of less than 1000 euros. Therefore, we introduce a relevance vector for each attribute to express the relation between attributes' values and queries. Let $W_q^{a_i} = [w_1 \ldots w_n]$ be the relevance vector of query q of attribute a_i. For any $u \in \mathcal{U}$ and $q \in \mathcal{Q}$, the relevance value of user u's profile to query q can be calculated as follows:

$$v_u(q) = \sum_{i\leq m} \vec{\Phi}_u^{a_i} \cdot [W_q^{a_i}]^T \tag{4.6}$$

where $[W_q^{a_i}]^T$ is the transpose of $W_q^{a_i}$. Suppose the relevance vector of attribute salary to a five-star hotel is $[0\ 0.2\ 0.6]$ then $v_u(q) = [0\ 1\ 0] \cdot [0\ 0.2\ 0.6]^T = 0.2$.

Finally, we can calculated u's *a priori* probability $\Pr_u(q|\mathcal{C}_t)$ as follows:

$$\Pr_u(q|\mathcal{C}_t) = \frac{v_u(q)}{\sum_{q' \in \mathcal{Q}} v_u(q')}. \tag{4.7}$$

As users are independent from each other to decide next queries to issue and user profiles are the only additional information in $\mathcal{C}_t$ to the inherent contexts, for the sake of simplicity we use $\Pr_u(q|\mathcal{P}_u)$ to replace $\Pr_u(q|\mathcal{C}_t)$ when there is no confusion from the context.

4.4 Privacy Analysis based on Query Dependency

In this section, we identify a new type of contextual information: *query dependency* and present how to incorporate it into our framework.

Since the first commercial LBSs launched in 1996, LBSs have evolved from simple single-target finder to diverse, proactive and multi-target services [BKH08]. However, due to the lack of user privacy protection, especially at the beginning, LBS providers accumulate a large amount of users' historical requests. What makes the situation worse is the shift of LBS providers from telecom operators (who were viewed as trusted entities) to open businesses such as Google Latitude, Foursquare, and MyTracks. This increases the risk of potential misuse of the accumulated records due to the new sensitive information derived from them.

The dependency between queries is one type of such sensitive but personal information. It is contained in users' requests because of users' preference in arranging their daily activities [GHB08]. This preference leads to a repetitive pattern in their requests. For instance, a user often posts a check-in of a coffee house after lunch. The fact that users' frequent queries are usually restricted to a small set makes the extraction of query dependency more precise.

Users' query dependency can be abused and becomes a potential risk to users' query privacy. As far as we know, we are the first to explore *query dependency* for query privacy protection. We illustrate this by a simple example.

Example 4.1. *Bob issues a request about the nearest night clubs in a* 2*-anonymous region with Alice being the other user. Suppose the adversary has also learnt that Alice just issued a query about the nearest clinics and Bob queried about bars. As it is not common to ask clubs after clinics compared to bars, the adversary can infer that Bob is more probable to issue the request about night clubs. In this example, even if Alice and Bob share a similar profile, the dependency between queries obviously breaks* 2*-anonymity for all users in the region who are supposed to be equally likely to issue the request.*

In the rest of this section, we start with a formal definition of the adversary's knowledge and then give an approach to derive dependency between queries for a user from his request history. Then we propose a method for the adversary to breach users' query privacy by exploring query dependency.

4.4.1 Updating the adversary's knowledge

Besides the contextual information considered in Section 4.3, there are two new types of contextual information added: *request history* and *observed request traces*.

As we have mentioned before, LBS providers have collected users' request history. For each user u, we assume that the adversary has a user u's request history for a sufficiently long period. We use a sequence to denote the requests of user u collected by the adversary, i.e., $\mathcal{H}_u = (\langle u, \ell_1, t_1, q_1\rangle, \ldots, \langle u, \ell_n, t_n, q_n\rangle)$ $(\forall_{1\leq i\leq n-1} t_i < t_{i+1})$. The ith request in $\mathcal{H}_u$ is represented by $\mathcal{H}_u(i)$. We call this sequence *user request history*, whose length is denoted as $len(\mathcal{H}_u)$. For the sake of simplicity, we assume that $\mathcal{H}_u$ is complete, namely there do not exist any requests that are issued by u during the period but are not included in $\mathcal{H}_u$.

Another assumption is that the adversary has access to the public channel which transmits generalised requests. This means that the adversary can obtain any generalised requests from users. We denote this contextual information by a sequence of generalised requests in the chronologically ascending order. Up to time t, the sequence of observed requests is $\mathcal{O}_t = (\langle r_1, t_1, q_1\rangle, \ldots, \langle r_n, t_n, q_n\rangle)$ $(\forall_{1\leq i\leq n} t_i < t_{i+1}$ and $t_n < t)$. For the sake of simplicity, we do not consider recurrent queries, i.e., those elements in $\mathcal{O}_t$ with the same time-stamps. Furthermore, for each request in $\mathcal{O}_t$, the adversary calculates its anonymity set, i.e., the users located in the generalised region. Thus, for each user, the adversary can maintain a sequence of generalised requests, whose anonymity sets contain this user. We call this sequence an *observed request trace* and denote the one of user u up to t as $\mathcal{O}_{u,t}$ whose length is $len(\mathcal{O}_{u,t})$. Obviously with time passing, a user's observed request trace keeps growing. The difference between $\mathcal{H}_u$ and $\mathcal{O}_{u,t}$ is that the adversary knows the issuer of each request in $\mathcal{H}_u$ but uncertain about the issuers of the requests in $\mathcal{O}_{u,t}$.

To summarise, the knowledge of the adversary can be formulated as the following:

$$\mathcal{C}_t = \{\mathcal{D}_t, f, \{\Phi_u \,|\, u \in \mathcal{U}\}, \mathcal{O}_t, \{\mathcal{H}_u \,|\, u \in \mathcal{U}\}\}. \tag{4.8}$$

4.4.2 Deriving query dependency

Query dependency can be used to predict the next query of a user based on his past queries. However, when a user has no past queries or the past queries have little impact on his future queries, we need to consider users' *a priori preference* on queries.

Query dependency. We model query dependency with the assumption that the query that a user will issue next can only be affected by the last query that the user has issued (i.e., the Markov property). For a pair of queries q_i and q_j, the dependency of query q_j on q_i can thus be expressed by the conditional probability $\Pr_u(q_j \,|\, q_i)$.

To find dependent queries, we need to identify the successive requests. Intuitively, two requests are successive if there are no other requests between them in the request history. This simply means that $\mathcal{H}_u(i+1)$ is the successive request of $\mathcal{H}_u(i)$ for $i < len(\mathcal{H}_u)$. All the occurrence of query q_j depending on q_i can be captured by the set of successive request pairs $\mathcal{S}_{i,j} = \{(\mathcal{H}_u(k), \mathcal{H}_u(k+1)) \mid req(\mathcal{H}_u(k)) =$

$q_i \wedge req(\mathcal{H}_u(k+1)) = q_j, 0 < k < len(\mathcal{H}_u)\}$. Given a request history $\mathcal{H}_u$, the adversary can derive for the user u his dependency between any pair of queries based on the sets $\mathcal{S}_{i,j}$. In this chapter we make use of Lidstone's or additive smoothing [MS99] to ensure that there is no dependency of degree zero for q_j on q_i due to no occurrence of the pair (q_i, q_j) in the request history. Formally, let λ be the smoothing parameter which is usually set to 1. The dependency $\Pr_u(q_j \,|\, q_i)$ is calculated as follows:

$$\Pr_u(q_j \,|\, q_i) = \frac{|\,\mathcal{S}_{i,j}\,| + \lambda}{\sum_{q_k \in \mathcal{Q}} |\,\mathcal{S}_{i,k}\,| + |\,\mathcal{Q}\,| \cdot \lambda}. \tag{4.9}$$

***A priori* preference.** There are many cases that a query does not depend on its past queries. For example, users may issue an LBS query for the first time or accidentally for an emergent need. In such cases, the best the adversary can do is to apply users' *a priori* preference to find the possible issuer.

We model the *a priori* preference of a user u as a distribution over the set of queries indicating the probability of the user to issue a query. For query $q_i \in \mathcal{Q}$, we denote by $\Pr_u(q_i)$ the probability that user u issues the query q_i when there is no dependence on any previous queries. It is obvious that $\sum_{q_i \in \mathcal{Q}} \Pr_u(q_i) = 1$.

There are many sources of information reflecting users' *a priori* preference. Users' personal information, i.e., user profiles, have been discussed in Section 4.3 and shown effective in assessing users' preference on queries [SAV08, SAV11]. Moreover, a user's request history also reflects his preference. Thus, we estimate a user's *a priori* preference by combining his request history ($\mathcal{H}_u$) and his user profile. Recall that we calculate a distribution for each user over the set of queries indicating the probability that the user issues a query based on his profile, i.e., $\Pr_u(q_i \mid \mathcal{P}_u)$. Moreover, let $\Pr_u(q_i \mid \mathcal{H}_u)$ be the likelihood for user u to issue q_i based on his request history. We can use the frequency of the occurrence of the query in the request history to estimate $\Pr_u(q_i \,|\, \mathcal{H}_u)$:

$$\Pr_u(q_i \,|\, \mathcal{H}_u) = \frac{|\,\{\mathcal{H}_u(k) \,|\, query(\mathcal{H}_u(k)) = q_i\}\,|}{len(\mathcal{H}_u)}. \tag{4.10}$$

The two distributions evaluate a user's *a priori* preference on next queries from two different perspectives. An agreement between them is needed. This is equivalent to aggregate expert probability judgements [BW93]. We use *linear opinion pool aggregation* which is empirically effective and has been widely applied in practice [AAB+00]. By assigning a weight to each distribution, i.e., $w_{\mathcal{P}}$ and $w_{\mathcal{H}}$ with $w_{\mathcal{P}} + w_{\mathcal{H}} = 1$, we can calculate $\Pr_u(q_i)$ as follows:

$$\Pr_u(q_i) = w_{\mathcal{P}} \cdot p_u(q_i \,|\, \mathcal{P}_u) + w_{\mathcal{H}} \cdot \Pr_u(q_i \,|\, \mathcal{H}_u). \tag{4.11}$$

Remark. The way we model users' query dependency and *a priori* preference has some restrictions. For instance, we do not consider the influence of factors such as the time when LBS requests are issued: usually a user's behaviours on weekdays are different from weekends [CPX14]. By distinguishing the request history at different time periods, the impact of time can be taken into account. We have also assumed that a query is only dependent on its immediate previous query. This

restriction can be lifted by considering, e.g., the last k historical queries. However, such dependency might not be as efficient and accurate as the probabilities of the form of $\Pr_u(q_i \mid q_j)$. An interesting factor is the time intervals between successive requests, which may present certain patterns as well. For instance, a user may prefer to issue a request within a specific amount of time after the previous one. This leads to various probabilities for a user to issue a query when he chooses different issuing time. In Section 4.4.4, we take time intervals between requests as an example to illustrate how to extend our model of query dependency to capture more influencing factors.

4.4.3 Query privacy analysis

Recall that the purpose of the adversary is to calculate the distribution $\Pr(U = u \mid \langle r, t, q\rangle, \mathcal{C}_t)$ given a generalised request $\langle r, t, q\rangle$. In the adversary's knowledge, the observed request list ($\mathcal{O}_t$) is the only dynamic context besides the spatial distribution (i.e., $\mathcal{D}_t$) which is inherently contained. For the sake of simplicity, we use $\Pr(u \mid \langle r, t, q\rangle, \mathcal{O}_t)$ for short to represent $\Pr(U = u \mid \langle r, t, q\rangle, \mathcal{C}_t)$ whenever no confusion exists.

The key of query privacy analysis is still to calculate $\Pr_u(q \mid \mathcal{C}_t)$ (i.e., $\Pr_u(q \mid \mathcal{O}_t)$ for short). Due to the independence between users with respect to issuing requests, a user's requests have no influence on the next query of any other user. Thus, $\Pr_u(q \mid \mathcal{O}_t) = \Pr_u(q \mid \mathcal{O}_{u,t})$.

The size of $\mathcal{O}_{u,t}$ is an important factor determining the accuracy and the complexity of the calculation of $\Pr_u(q \mid \mathcal{O}_{u,t})$. Recall that $\mathcal{O}_{u,t}$ consists of all the observed requests that may be issued by user u up to time t. Intuitively, the longer $\mathcal{O}_{u,t}$ is, the more computational overhead is required to obtain $\Pr(u \mid \langle r, t, q\rangle, \mathcal{O}_t)$. Therefore, it is not practical to consider the complete $\mathcal{O}_{u,t}$. Instead, we fix a *history window* which consists of the latest n observed requests of user u ($n \leq len(\mathcal{O}_{u,t})$). Our problem can thus be reformulated as to compute $\Pr^n(u \mid \langle r, t, q\rangle, \mathcal{O}_t)$, indicating that the distribution is based on the last n observed requests.

In Figure 4.3, we show an example of a history window which contains n observed requests, $\langle r_{i_1}, t_{i_1}, q_{i_1}\rangle, \ldots, \langle r_{i_n}, t_{i_n}, q_{i_n}\rangle$ with $t_{i_j} > t_{i_{j-1}}$ ($j > 1$). Let $\ell q_j(\mathcal{O}_{u,t})$ be the jth latest observed request in $\mathcal{O}_{u,t}$, whose query is $query(\ell q_j(\mathcal{O}_{u,t})) = q_{i_j}$. In the following discussion, we simply write ℓq_j if $\mathcal{O}_{u,t}$ is clear from the context. It is obvious that ℓq_1 is the latest observed request of user u.

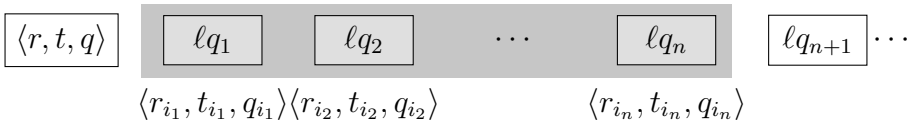


Figure 4.3: A history window of n observed requests.

Once $\Pr^n(u \mid \langle r, t, q\rangle, \mathcal{O}_t)$ is calculated, it is added into the adversary's knowledge. Therefore, for a past request $\langle r', t', q'\rangle$ in $\mathcal{O}_{u,t}$ ($t' < t$), the adversary has $\Pr(u \mid \langle r', t', q'\rangle, \mathcal{O}'_t)$. In the sequel, we simply denote it as $\Pr(u \mid \langle r', t', q'\rangle)$ in cases without confusion.

A user's latest request determines the probability distribution of his next query. Whereas, it is uncertain which is the latest in the history window. To handle this uncertainty, we distinguish three cases which are depicted in Figure 4.4.

1. User u has issued both the last request (i.e., ℓq_1, see Figure 4.4(a)) and the current request (i.e., $\langle r, t, q\rangle$). Considering query dependence, the probability of this case is
$$\Pr_u(u \,|\, \ell q_1) \cdot p_u(q \,|\, q_{i_1}). \tag{4.12}$$

2. User u has issued the current request $\langle r, t, q\rangle$ and his latest request is ℓq_m $(1 < m \leq n)$ (see Figure 4.4(b)). The probability of ℓq_m being the latest request is the production of the probability that the last $m-1$ requests are *not* issued by u and the probability that u has issued ℓq_m, i.e., $\Pr(u \,|\, \ell q_m) \cdot \prod_{j=1}^{m-1}(1 - \Pr(u \,|\, \ell q_j))$. Considering query dependence, the probability of this case is
$$\Pr_u(q \,|\, q_{i_m}) \cdot \Pr(u \,|\, \ell q_m) \cdot \prod_{j=1}^{m-1}(1 - \Pr(u \,|\, \ell q_j)). \tag{4.13}$$

3. User u did not issue any request in the history window (see Figure 4.4(c)). In this case, we suppose that the user issued the current request according to his *a priori* preference, i.e., $\Pr_u(q)$. Based on the probability that the user's latest request is outside of the history window as $\prod_{j=1}^{n}(1 - \Pr(u \,|\, \ell q_j))$, the probability of this case is
$$\Pr_u(q) \cdot \prod_{j=1}^{n}(1 - \Pr(u \,|\, \ell q_j)). \tag{4.14}$$

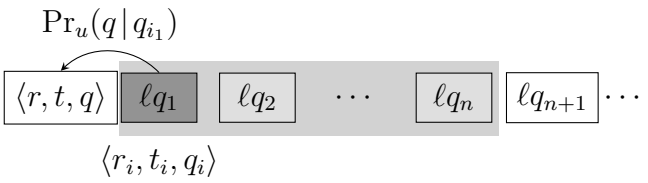
We sum up the above three probabilities to compute the probability for user u in region r at time t to issue q when a history window of size n is considered:

$$\begin{aligned}\Pr_u^n(q \,|\, \mathcal{O}_{u,t}) = \;& \Pr(u \,|\, \ell q_1) \cdot \Pr_u(q \,|\, query(\ell q_1)) \\ & + \sum_{m=2}^{n} \Pr(u \,|\, \ell q_m) \cdot \Pr_u(q \,|\, query(\ell q_m)) \cdot \prod_{j=1}^{m-1}(1 - \Pr(u \,|\, \ell q_j)) \\ & + \Pr_u(q) \cdot \prod_{j=1}^{n}(1 - \Pr(u \,|\, \ell q_j)).\end{aligned} \tag{4.15}$$

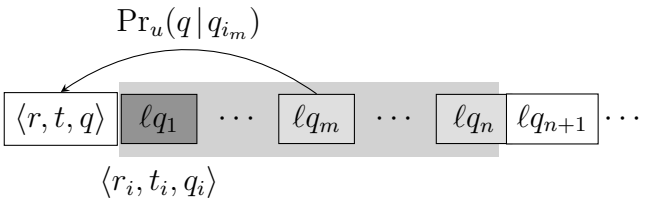
We use the following example with $n = 2$ to show the calculation.

Example 4.2. *Suppose the last two requests are $\langle r'', t'', q''\rangle$ and $\langle r', t', q'\rangle$ with $t'' < t' < t$ in $\mathcal{O}_{u,t}$. Let $\langle r, t, q\rangle$ be an observed request. Then for user u, the probability that he issues the request is computed as follows:*

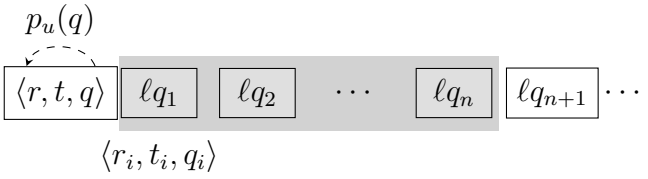
$$\begin{aligned}Pr_u^2(q \,|\, \mathcal{O}_{u,t}) = \;& Pr_u(q \,|\, q') \cdot Pr(u \,|\, \langle r', t', q'\rangle) \\ & + \big(1 - Pr(u \,|\, \langle r', t', q'\rangle)\big) \cdot Pr(u \,|\, \langle r'', t'', q''\rangle) \cdot Pr_u(q \,|\, q'') \\ & + \big(1 - Pr(u \,|\, \langle r', t', q'\rangle)\big) \cdot \big(1 - Pr(u \,|\, \langle r'', t'', q''\rangle)\big) \cdot Pr_u(q).\end{aligned}$$



(a) The latest request is ℓq_1.



(b) The latest request is ℓq_m $(1 < m \leq n)$.



(c) The latest request is not in the history window.

Figure 4.4: The three cases.

4.4.4 Handling the time intervals between requests

In this section, we study a factor that has impact on query dependency: time intervals between two successive requests.

It has been noted that not only the behaviours of a user follow certain patterns but also the amount of time between behaviours. For instance, Giannotti et al. [GNPP06, GNPP07] study and extract the pattern of the time intervals between events in sequential pattern mining. The idea is also adopted by Chen et al. [CPX13] for constructing and comparing user mobility profiles. Similarly, with respect to LBS requests, users can also have their preferences on the time intervals between two successive requests.

Example 4.3. *Consider a user who is in a city centre and wants to meet his friends at a bar. He first sends an LBS request asking for nearby friends who can potentially meet together. Then the user contacts those friends and discuss with them about their availability, which takes about half an hour. Afterwards, he issues another request for nearby bars.*

In the above example, the time interval between the two requests should usually be around 30 minutes. Suppose that the user sent another query two minutes after the first query about nearby friends. Then this query is less likely to be a query on nearby bars, compared to the situation when a query is issued about 30 minutes later. Therefore, query dependency should vary according to when the next query is issued.

To capture the influence of query issuing time, given two queries q_i and q_j, instead

of $\Pr_u(q_j \,|\, q_i)$ we calculate the distribution $\Pr_u(q_j \,|\, q_i, \tau)$, where τ is the amount of time after user u issued the last request with query q_i. This distribution can be calculated based on other distributions deduced from the following equation:

$$\begin{aligned}\Pr_u(q_j \,|\, q_i, \tau) =& \frac{\Pr_u(\tau \,|\, q_j, q_i) \cdot \Pr_u(q_j, q_i)}{\Pr_u(\tau \,|\, q_i) \cdot \Pr_u(q_i)} \\ =& \frac{\Pr_u(\tau \,|\, q_j, q_i)}{\Pr_u(\tau \,|\, q_i)} \cdot \Pr_u(q_j \,|\, q_i).\end{aligned} \tag{4.16}$$

There are two new distributions in the above equation. The first one is $\Pr_u(\tau \,|\, q)$ indicating the probability that a user issues a successive query with time interval τ after issuing query $q \in \mathcal{Q}$. The other distribution is $\Pr_u(\tau \,|\, q_j, q_i)$ meaning the likelihood that if user u issues query q_j after q_i, then time interval between them is τ.

The time interval between requests can be considered as a random variable T. The above two distributions can thus be calculated based on the probability density functions of T in different cases, i.e., $\hat{f}(T \mid q)$ and $\hat{f}(T \mid q_j, q_i)$. Let ϵ be the granularity of time, e.g., a second or a minute. Then given a time interval τ, we have the following calculation:

$$\Pr_u(\tau \,|\, q_i) = \int_{\tau}^{\tau+\epsilon} \hat{f}(T \,|\, q_i) \mathrm{d}T; \tag{4.17}$$

$$\Pr_u(\tau \,|\, q_j, q_i) = \int_{\tau}^{\tau+\epsilon} \hat{f}(T \,|\, q_j, q_i) \mathrm{d}T. \tag{4.18}$$

The problem of density estimation based on observed data has been extensively studied and some classic methods have been developed in practice, e.g., the kernel smoothing estimator [DL01]. In our case, the key to estimate the density function of T is to extract the corresponding set of observed samples of time intervals. Take $\hat{f}(T \,|\, q_j, q_i)$ as an example. The samples of time intervals form a multi-set which can be obtained from users' request history, e.g., $\mathcal{H}_u$. Recall that $\mathcal{S}_{i,j}$ is the set of pairs of successive requests whose queries are q_i and q_j, respectively. Then the observed set of time intervals is

$$\{t' - t \,|\, (\langle r, t, q_i \rangle, \langle r', t', q_j \rangle) \in \mathcal{S}_{i,j}\}.$$

The calculation of user u's *a priori* probability at time t to issue query q (i.e., Equation 4.15) can thus be extended to handle the query dependency with respect to time intervals. The calculation is shown in Equation 4.19:

$$\begin{aligned}& \Pr_u^n(q \,|\, \mathcal{O}_{u,t}) \\ =\ & \Pr(u \,|\, \ell q_1) \cdot \Pr_u(q \,|\, \mathit{query}(\ell q_1), t - \mathit{time}(\ell q_1)) \\ & + \sum_{m=2}^{n} \Pr(u \,|\, \ell q_m) \cdot \Pr_u(q \,|\, \mathit{query}(\ell q_m), t - \mathit{time}(\ell q_m)) \cdot \prod_{j=1}^{m-1} (1 - \Pr(u \,|\, \ell q_j)) \\ & + \Pr_u(q) \cdot \prod_{j=1}^{n} (1 - \Pr(u \,|\, \ell q_j)).\end{aligned} \tag{4.19}$$

4.5 Measuring Query Privacy

LBS requests are generalised to protect the issuers' query privacy. The level of query privacy offered by the generalisation algorithms should be quantified precisely. This is due to (i) the generalisation algorithm requires the evaluation so as to improve their performance; (ii) LBS users need the quantification to express their privacy requirements for their requests.

Besides k-anonymity, many privacy metrics have been proposed in the literature, such as correctness-based [STBH11], estimation error-based [RMPADF13] and feeling-based [XC09]. These metrics quantify query privacy from different perspectives. For instance, the feeling-based metric makes use of entropy to evaluate the average uncertainty of the adversary to guess the issuer in a given scenario (e.g., shopping mall) which is subsequently used as the privacy requirement of users. Correctness-based metrics quantify privacy as the probability of the adversary choosing the right issuer when he makes a single guess. Using our framework, we can adopt the ideas of these metrics, which leads to a diverse and comprehensive series of measurements for query privacy. In this section, we present three new metrics on query privacy and formally define them using our framework.

Inspired by anonymity degrees defined by Reiter and Rubin [RR98], we come up with the following two new privacy metrics: *k-approximate beyond suspicion* and *user specified innocence*. Note that user specified innocence coincides with the idea of correctness-based metrics. Furthermore, we propose a third metric by using entropy.

k-approximate beyond suspicion. *Beyond suspicion* means from the attacker's viewpoint, the issuer cannot be more likely than other potential users in the anonymity set to issue the query. In the context of LBSs, we need to find a set of users in which users are the same likely to send a given query. This set is taken as the anonymity set whose size determines the degree of users' privacy as in k-anonymity. Let $AS : Q' \rightarrow 2^{\mathcal{U}}$ denote the anonymity set of a generalised request. The issuer of query $\langle u, whereis(u,t), t, q\rangle$ is beyond suspicious with respect to the corresponding generalised request $\langle r, t, q\rangle$ if and only if $\forall u' \in AS(\langle r, t, q\rangle)$,

$$\Pr(u \,|\, \langle r, t, q\rangle, \mathcal{C}_t) = \Pr(u' \,|\, \langle r, t, q\rangle, \mathcal{C}_t). \tag{4.20}$$

In practice, the number of users with the same probability to send a query is usually small, which leads to a large generalised area with a fixed k. So we relax the requirement to compute an anonymity set consisting of users with *similar probabilities* instead of the exact same probability. Let $\|\Pr_1, \Pr_2\|$ denote the difference between two probabilities and ϵ be the pre-defined parameter describing the largest difference allowed between similar probabilities.

Definition 4.2 (k-approximate beyond suspicion)**.** *Let $\langle u, whereis(u,t), t, q\rangle \in Q$ be a query and $\langle r, t, q\rangle \in Q'$ the corresponding generalised request. The issuer u is* k-approximate beyond suspicious *if*

$$|\, \{u' \in AS(\langle r, t, q\rangle) \mid \|Pr(u \,|\, \langle r, t, q\rangle, \mathcal{C}_t), Pr(u' \,|\, \langle r, t, q\rangle, \mathcal{C}_t)\| < \epsilon \,| \geq k.$$

Different from k-anonymity, the set of users that are k-approximate beyond suspicious is computed based on the spatial distribution of users with similar probabilities rather than the original distribution involving all users. The users in an

anonymity set have similar probabilities and the size of the anonymity set is larger than k. Therefore, k-approximate beyond suspicion can be seen as a generalised version of k-anonymity. If for a specific query $q \in \mathcal{Q}$, any two users have the same probability to issue it, then k-approximate beyond suspicion is equivalent to k-anonymity. For short, we use k-ABS to denote k-approximate beyond suspicion in the following discussion.

User specified innocence. *Probable innocence* and *possible innocence* are proposed by Reiter and Rubin [RR98]. An issuer is probably innocent if from the attacker's view the issuer appears no more likely to be the originator of the query. In other words, the probability of each user in the anonymity set to be issuer should be less than 50%. Meantime, possible innocence requires the attacker not be able to identify the issuer with a non-trivial probability. We extend these two notions into a metric with user-specified probabilities (instead of restricting to 50% or non-trivial probability which is not clearly defined). We call the new anonymity metric *user specified innocence* where $\alpha \in [0, 1]$ is the specified probability given by the issuer. Intuitively, for a query, an issuer is α-user specified innocent, if the anonymiser generates the same region for any user in the region with the same specified value α. In other words, in the generalised region, the most probable user has a probability smaller than α. Recall that $u\ell(r, t)$ denotes the set of users in region r at time t. It is clear that the anonymity set consists of all users in the generalised area.

Definition 4.3 (User specified innocence)**.** *Let $\alpha \in [0, 1]$, $\langle u, whereis(u, t), t, q\rangle \in Q$ be a query and $\langle r, t, q\rangle \in Q'$ the corresponding generalised request. The issuer u is* α-user specified innocent *if for all $u' \in u\ell(r, t)$,*

$$Pr(u' \,|\, \langle r, t, q\rangle, \mathcal{C}_t) \leq \alpha.$$

We abbreviate α-user specified innocence as α-USI.

An entropy-based metric. Serjantov and Danezis [SD03] define an anonymity metric based on entropy and Díaz et al. [DSCP03] provide a similar metric that is normalised by the number of users in the anonymity set. The concept *entropy* of a random variable X is defined as $H(X) = -\sum_{x\in\mathcal{X}} p(x) \cdot \log p(x)$ where $\mathcal{X}$ is the domain (all possible values) of X. In our context, entropy can also be used to describe the attacker's uncertainty to identify the issuer of a generalised request. Let variable U denote the issuer of query $\langle r, t, q\rangle$. Then the uncertainty of the attacker can be expressed as

$$H(U \,|\, \langle r, t, q\rangle, \mathcal{C}_t) = -\sum_{u'\in u\ell(r,t)} \Pr(u' \,|\, \langle r, t, q\rangle, \mathcal{C}_t) \cdot \log \Pr(u' \,|\, \langle r, t, q\rangle, \mathcal{C}_t). \quad (4.21)$$

For a given generalised request $\langle r, t, q\rangle$ and a given value β, we say that the issuer is entropy-based anonymous with respect to the value β if all users in region r can have r as the generalised region when issuing the same query and the entropy $H(U \,|\, \langle r, t, q\rangle, \mathcal{C}_t)$ is not smaller than β.

Definition 4.4 (Entropy-based anonymity)**.** *Let $\beta > 0$, $\langle u, whereis(u, t), t, q\rangle \in Q$ be a query and $\langle r, t, q\rangle \in Q'$ the corresponding generalised request. The issuer u is* β-entropy based anonymous *if*

$$H(U \,|\, \langle r, t, q\rangle, \mathcal{C}_t) \geq \beta.$$

For short, we call β-entropy based anonymity β-EBA.

Remark. When users use these metrics to express their privacy requirements, at least three elements should be provided: a metric, the values of the parameters required by the chosen metric (e.g., k, α), and the values of the parameters used to calculation posterior probabilities (e.g., the size of history windows).

In practice it is difficult and cumbersome for a user to give exact values to the elements. First, all the metric values in requirements should be determined before requests are generalised (i.e., ex-ante) but they are defined ex-post in nature in the metric. Furthermore, users need to understand the meaning of each parameter and the corresponding implication on privacy protection. To avoid this situation, in this chapter we provide a list of privacy levels, e.g., from *low* to *very high*. Each level corresponds to a setting of privacy parameters. For example, when query dependency is considered, a user's privacy requirement can be represented as $\langle \mathsf{kABS}, \mathit{high} \rangle$, which is then transformed into $\langle \mathsf{kABS}, (10, 0.05), (5) \rangle$. This ensures that whenever a request is successfully generalised, the region contains 10 users with similar posterior probabilities to the issuer's, after taking into account the last 5 observed requests. Furthermore, the distance between two such users' posterior probabilities is bounded by 0.05. In practice, the transformation can be made automatic and embedded in the request generalisation process. Note that the existing works can also be adapted to determine the values, e.g., the feeling-based privacy metric [XC09].

4.6 Generalisation Algorithms

In this section, we develop area generalisation algorithms to compute regions satisfying users' privacy requirements expressed in the proposed metrics in Section 4.5. As to find a region satisfying k-ABS is similar to compute a region satisfying k-anonymity on a given spatial distribution, we design an algorithm for k-ABS by combining the algorithm grid [MBFW07] with a clustering algorithm. For the other metrics, we design a uniform algorithm based on dichotomicPoints [MBFW07].

4.6.1 An algorithm for k-ABS

To find an area that satisfies k-ABS is to guarantee that at least k users in the area have similar posterior probabilities. This task can be divided into two main steps. The first is to obtain the spatial distribution of the users who have similar *a priori* probabilities to the issuer (e.g., $\Pr_u(q \,|\, \mathcal{C}_t)$). The second step is to run a k-anonymity generalisation algorithm to find a region with at least k users based on the spatial distribution computed at the first step.

The first step can be transformed to the clustering problem. Given $q \in \mathcal{Q}$, we need to cluster the users in $\mathcal{U}$ such that the users with similar *a priori* probabilities with respect to issuing q are grouped together.

For the second step, we use algorithm grid by Mascetti et al. [MBFW07] as it generates regular regions with smaller area compared to others. A two-dimensional space is partitioned into a grid with $\lfloor \frac{N}{k} \rfloor$ cells each of which contains at least k

users, where N denotes the number of users in $\mathcal{U}$. A user's position is represented by two dimensions x and y. The algorithm grid consists of two steps. First, users are ordered based on dimension x, and then on y. The ordered users are divided into $\lfloor\sqrt{\frac{N}{k}}\rfloor$ blocks of consecutive users. The block with the issuer enters the second step. The users in this block are then ordered first based on dimension y and then x. These users are also partitioned into $\lfloor\sqrt{\frac{N}{k}}\rfloor$ blocks. Then the block with the issuer is returned as the anonymity set. Details of the grid algorithm can be found in [MBFW07].

Algorithm 4.1 describes our algorithm for k-ABS. In general, it gives the generalised region as output which satisfies the user requirement k. Function cluster returns the cluster of users with similar probabilities to that of u with respect to query q. Then the function grid outputs a subset of *sim_users* with at least k users who are located in the rectangular region. The generalised region is computed by function region.

Algorithm 4.1 A generalisation algorithm for k-ABS.

1: FUNCTION: kABS
2: INPUT: $\langle u, whereis(u,t), t, q\rangle, dis(t), k, \mathcal{M}(q) = \{\Pr_{u'}(q\,|\,\mathcal{C}_t)\,|\,u' \in \mathcal{U}\}$
3: OUTPUT: A region r that satisfies k-ABS
4:
5: *sim_users* :=cluster$(u, q, \mathcal{M}(q))$;
6: AS := grid$(sim_users, dis(t), k)$;
7: r := region(AS)

Note that the clustering algorithm does not have to run each time when there is a request coming to the anonymiser. As long as the spatial distribution remains static or does not have big changes, for the requests received during this period, the anonymiser just executes the clustering algorithm once and returns the cluster containing the issuer as output of function cluster. The choice of the clustering algorithms has an impact on the performance of the generalisation algorithm. The complexity of Algorithm 4.1 is the sum of those of the clustering algorithm implemented and the gird algorithm ($\mathcal{O}(\sqrt{kN}\log\sqrt{kN})$ [MBFW07]).

4.6.2 An algorithm for α-USI and β-EBA

For privacy metrics α-USI and β-EBA, we design a uniform algorithm where users can specify which metric to use. Recall that in grid, the number of cells is predetermined by k and the number of users. Thus it is not suitable to perform area generalisation for metrics without a predefined number k. Instead we use algorithm dichotomicPoints.

The execution of dichotomicPoints involves multiple iterations in each of which users are split into two subsets. Similar to grid, positions are represented in two dimensions x and y, and users are also ordered based on their positions. However, different from grid the orders between axes are determined by the shape of intermediate regions rather than fixed beforehand. Specifically, if a region has a longer projection on dimension x, then x is used as the first order to sort the users.

Otherwise, y is used as the first order. Users are then ordered based on the values of their positions on the first order axis and then the second order. Subsequently, users are partitioned into two blocks with the same or similar number of users along the first order axis. The block containing the issuer is taken into the next iteration. This process is repeated until any of the two blocks contains less than $2k$ users. This termination criterion is to ensure security against the outlier problem (see Section 4.8).

However, in our uniform algorithm, instead of checking the number of users, we take the satisfaction of users' privacy requirement as the termination criterion, e.g., if all users in the two blocks have a probability smaller than α.

Given a request, our uniform algorithm executes three main steps to calculate the generalised region. The first step is to update users' *a priori* probabilities (at time t) based on the latest contextual information $\mathcal{C}_t$. This is done by the procedure updatePriori. This step can be skipped if the evolution of the contextual information does not affect the *a priori* probabilities, e.g., when only user profiles are contained. In the second step, after determining the first order axis, we call function updateAS to find a smaller anonymity set. It takes a set of users and partitions them into two subsets along the first order axis, both of which should satisfy the issuer's privacy requirement and updateAS returns the one containing the issuer as the updated anonymity set. When it is not possible to partition users along the first order axis, i.e., one of the two blocks generalised by any partition fails the issuer's requirement, the second order axis will be tried. If both tries have failed, updateAS simply returns the original set, which means no possible partition can be made with respect to the privacy requirement. In this situation, the whole algorithm terminates. Otherwise, the new set of users returned by updateAS is taken into the next iteration. Last, if the request can be generalised, then we should update the contextual information to include the generalised request, e.g., the observed request lists (i.e., $\mathcal{O}_t$, $\mathcal{O}_{u,t}$). This is done by calling the function updateContext whose implementation is determined by the exploited contextual information.

Algorithm 4.2 describes the uniform algorithm in detail. Function check(AS, $req(qu)$) calculates the normalised *a priori* probability of each user in AS. Then the function takes the resulted normalised probabilities as the users' posterior probabilities and check whether they satisfy the requirement $req(qu)$. The boolean variable *cont* is used to decide whether the algorithm should continue. It is set to false when the set of users in $\mathcal{U}$ does not satisfy the requirement (line 7) or when AS cannot be partitioned furthermore (line 30). The former case means that the requirement $req(qu)$ is set too high to be satisfied and the algorithm should immediately terminate while the latter case indicates that the generalised region is found. The anonymity set AS is represented as a two-dimensional array. After ordering users in AS, $AS[i]$ consists of all users whose positions have the same value on the first order axis. We use len($order$) to denote the size of AS in the dimension denoted by $order$. For instance, in Figure 4.5(a), axis x is the first order axis and $AS[3]$ has three users with the same x values. Moreover, len($first$) is 6.

The function updateAS shown in Algorithm 4.3 is critical for our algorithm uniformDP. It takes as input a set of users and outputs a subset that satisfies the issuer's privacy requirement $req(qu)$. It first orders the users and then divides them into

Algorithm 4.2 The uniform generalisation algorithm for α-USI and β-EBA.

```
1: FUNCTION: uniformDP
2: INPUT: qu=⟨u, ℓ, t, q⟩, req(qu), C_t
3: OUTPUT: Region r that satisfies req(qu)
4:
5: AS := U;
6: updatePriori(AS); \* for each u' ∈ AS, calculate Pr_u(q|C_t). *\
7: cont := check(AS, req(qu));
8: if cont = false then
9:     return ∅;
10: end if
11: while cont do
12:     min_x := min_{u'∈AS} whereis(u').x;
13:     min_y := min_{u'∈AS} whereis(u').y;
14:     max_x := max_{u'∈AS} whereis(u').x;
15:     max_y := max_{u'∈AS} whereis(u').y;
16:     if (max_x − min_x) ≥ (max_y − min_x) then
17:         first := x;
18:         second := y;
19:     else
20:         first := y;
21:         second := x;
22:     end if
23:     AS' = updateAS(AS, req(qu), first);
24:     if AS' = AS then
25:         AS' = updateAS(AS, req(qu), second);
26:     end if
27:     if AS' ≠ AS then
28:         cont := true;
29:     else
30:         cont := false;
31:     end if
32: end while
33:
34: updateContext(C_t);
35: return region(AS);
```

Algorithm 4.3 The function updateAS.

```
1: FUNCTION: updateAS
2: INPUT: AS, req(qu), order
3: OUTPUT: AS' ⊆ AS that contains u and satisfies req(qu)
4:
5: AS := reorder(AS, order);
6: i := mid(AS, order);
7: if check(left(i), req(qu)) ∧ check(right(i), req(qu)) then
8:     AS := part(i, u);
9: else
10:    found := false;
11:    j := 0;
12:    while j ≤ len(order) ∧ ¬found do
13:        if check(left(j), req(qu)) ∧ check(right(j), req(qu)) then
14:            found := true;
15:            AS := part(j, u);
16:        else
17:            j := j + 1;
18:        end if
19:    end while
20: end if
21: return AS;
```

two subsets with the same number of users along the first order axis (indicated by the variable *order*). This operation is implemented by the function $\mathsf{mid}(AS, order)$ which returns the middle user's index in the first dimension of AS. If both of the two subsets satisfy $req(qu)$, then the one containing the issuer is returned (implemented by function $\mathsf{part}(i, u)$). Otherwise, an iterative process is started. In jth iteration, the users are partitioned into two sets one of which contains the users in $AS[1], \ldots, AS[j]$ (denoted by $\mathsf{left}(j)$) and the other contains the rest (denoted by $\mathsf{right}(j)$). These two sets are checked against the privacy requirement $req(qu)$. If both $\mathsf{left}(j)$ and $\mathsf{right}(j)$ satisfy $req(qu)$, the one with issuer u is returned by $\mathsf{part}(j, u)$. If there are no partitions feasible after $\mathsf{len}(order)$ iterations, the original set of users is returned.

An example execution of Algorithm 4.2 is shown in Figure 4.5. The issuer is represented as a black dot. In Figure 4.5(a) the users are first partitioned into two parts from the middle. Assume both parts satisfy $req(qu)$, so the set b_1 is returned as the anonymity set AS for the next iteration. As b_1's projection on axis y is longer, the first order is set to axis y (Figure 4.5(b)). If after dividing the users from the middle, the set b_2 does not satisfy $req(qu)$. Thus, the users are partitioned from $AS[1]$ to $AS[4]$ (Figure 4.5(c)). Suppose no partitions are feasible. The first order axis is then switched to axis x. Function updateAS is called again to find a partition along axis x (Figure 4.5(d)).

We can see Algorithm 4.2 iterates for a number of times. In each iteration, some users are removed from the previous anonymity set. Operations such as partition and requirement check are time-linear in the size of the anonymity set. The number of iterations is logarithmic in the number of the users. So in the worst case, the

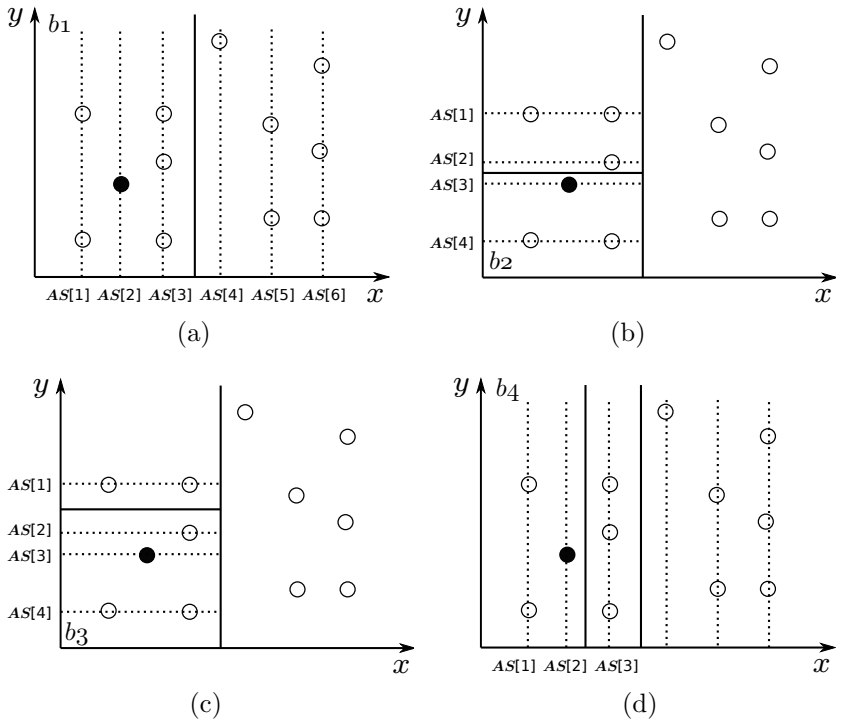


Figure 4.5: An example execution of our algorithm uniformDP.

time complexity of Algorithm 4.2 is $\mathcal{O}(N \log N)$, where N denotes the number of all users in $\mathcal{U}$. The correctness of Algorithm 4.2 is stated as Thm. 4.1.

Theorem 4.1. *For any query $\langle u, \ell, t, q\rangle$, Algorithm 4.2 computes a generalised region that satisfies the issuer u's privacy requirement $req(\langle u, whereis(u,t), t, q\rangle)$.*

Proof. By Definition 4.3 and Definition 4.4, Algorithm 4.2 computes a region r for a query $\langle u, whereis(u,t), t, q\rangle$ that satisfies a constraint related to the issuer's posterior probability and the entropy about the issuer. We take α-USI as an example to show the correctness of our algorithm and the proofs of the other two are analogous.

By Definition 4.3, we have to prove the posterior probability of each user $u' \in u\ell(r,t)$ is smaller than α, i.e., $\Pr(u' \,|\, \langle r, t, q\rangle, \mathcal{C}_t) \leq \alpha$. According to Equation 4.2 and Equation 4.3, we need to prove for any $u' \in u\ell(r,t)$ (1) $f(\langle u', whereis(u',t), t, q\rangle) = \langle r, t, q\rangle$ and (2) his normalised *a priori* probability over those of all users in region r should be smaller than α, i.e.,

$$\frac{\Pr(q \,|\, u', \mathcal{C}_t)}{\sum_{u'' \in u\ell(r,t)} \Pr(q \,|\, u'', \mathcal{C}_t)} \leq \alpha. \tag{4.22}$$

Let u' be any user in the generalised region r of Algorithm 4.2. Let AS_j and AS'_j be the values of AS in the jth iteration of Algorithm 4.2 of u and u', respectively. To prove (1), we show that $AS_j = AS'_j$ by induction on the number of iterations, i.e., j.

Induction basis: Initially, we suppose that $\mathcal{U}$ satisfied the requirement. Then we have $AS_1 = AS'_1$.

INDUCTION STEP: Assume at jth iteration $AS_j = AS'_j$. We have to show that the algorithm either terminates with AS_j and AS'_j, or enters the next iteration with $AS_{j+1} = AS'_{j+1}$. The equality that $AS_j = AS'_j$ is followed by that $\mathsf{mid}(AS_j, order) = \mathsf{mid}(AS'_j, order)$. There are three possible executions.

> Case 1: if $\mathsf{left}(i)$ and $\mathsf{right}(i)$ of AS_j and AS'_j satisfy the requirements (line 7 of Algorithm 4.3), the part containing the issuer is returned. Thus AS_{j+1} contains u as well as all other users in $u\ell(r,t)$, including u'. Thus, $AS_{j+1} = AS'_{j+1}$.
>
> Case 2: if the check at line 7 of Algorithm 4.3 fails, then the algorithm switches to find from the beginning the first feasible partition. Suppose the partition is made at the position x for AS_j. Then x is also the right position for AS'_j as $AS_j = AS'_j$. Because of the similar reason in the previous possible execution, the same subset is set to AS_{j+1} and AS'_{j+1}. Thus, $AS_{j+1} = AS'_{j+1}$.
>
> Case 3: if there are no possible partitions, Algorithm 4.3 returns AS_{j+1} and AS'_{j+1} in both cases. Then the first order is changed and Algorithm 4.3 is called again. If one of the first two execution is taken, with the analysis above, we have $AS_{j+1} = AS'_{j+1}$. Otherwise, Algorithm 4.2 terminates with $\mathsf{region}(AS_j)$ and $\mathsf{region}(AS'_j)$ which are equal.

We proceed with (2). Recall that the function $\mathsf{check}(AS, req(qu))$ returns true for metric α-USI only if Equation 4.22 holds for each user in AS because it takes users' normalised *a priori* probabilities as their posterior probabilities. At the line 5 of Algorithm 4.2, we set AS to the original user set $\mathcal{U}$ and the algorithm continues only if the function $\mathsf{check}(\mathcal{U}, req(qu)$ returns true. Otherwise, it is impossible to return a region satisfying the requirement. The set AS is only reassigned to another set when a partition is made (line 8 or line 15 in Algorithm 4.3). For the two sets by the partition check all returns true and the one containing the issuer is assigned to AS. Thus, it is guaranteed that for each user $u' \in u\ell(r,t)$, Equation 4.22 holds. □

4.7 Experimental Results

We conduct experiments to evaluate our work from two aspects. First, we test the effectiveness of our framework in terms of the changes of issuers' posterior probabilities. In this way, we illustrate that users' personal profiles and request histories do cause privacy risks. Second, we implement our algorithms presented in Section 4.6 and with the experimental results we show and compare the characteristics of our new metrics proposed in Section 4.5.

To perform the experiments, we construct two sample datasets to simulate the spatial distributions of a collection of mobile users (*mobility dataset*) and their issued requests during movements (*request dataset*). We generate the mobility dataset using the moving object generator [Bri02] and it consists of the trajectories of $38,500$ users in a period with 50 discrete time points. We compose a series of request datasets corresponding to different numbers of *active users*. A user is called active if he subscribes certain LBSs and would issue requests during the period. Given a number of active users, we simulate a trace of requests for each of them

according to his query dependency and his *a priori* preference on queries. Note that throughout the experiments, we do not distinguish users' *a priori* preferences from the *a priori* probabilities computed based on user profiles. This is because they are both static and *a priori* probabilities have already been considered in the calculation of *a priori* preferences. We assume 6 types of queries for users to choose. This makes users' *a priori* preference around 17% on average. As we mentioned, our purpose is to evaluate the privacy risk incurred by contextual information and the effectiveness of the algorithms. Thus we assume that users' query dependency is available and generate it by a random procedure. Users' *a priori* preference is assessed in a similar way.

Our simulation is implemented with Java and run on a Linux laptop with 2.67 Ghz Intel Core (TM) and 4GB memory.

4.7.1 Impact of contextual information

We validate the effectiveness of our framework by checking if it can increase the likelihood of the adversary to correctly identify issuers by obtaining more contextual information. Given a generalised request, we can use the issuer's posterior probability as the measurement of the correctness of the adversary's attack on query privacy [STBH11]. If a type of contextual information can help breach users' query privacy, then issuers will have larger posterior probabilities than those computed without the information on average. The main idea of our validation is to check whether the framework can capture this increase.

In our experiments, we construct three attack scenarios where k-anonymity spatial generalisation is deployed. In the first scenario, the adversary only learns the inherent contextual information while in the other two scenarios, users' *a priori* preferences and request histories are added sequentially to the adversary's knowledge. We denote the corresponding contextual information by $\mathcal{C}_t^{\mathsf{basic}}$, $\mathcal{C}_t^{\mathsf{pf}}$ and $\mathcal{C}_t^{\mathsf{dep}}$, respectively.

We define *correctness increase ratio* (CIR), and use it to quantify the increase of issuers' posterior probabilities when more contextual information is explored. Specifically, it is computed as the ratio of the increase over the posterior probabilities calculated without considering the contextual information. In this chapter, we consider two CIRs, i.e., Δp_{pf} and Δp_{dep}. For a generalised request $\langle r, t, q\rangle$ issued by u, they can be calculated as follows:

$$\Delta p_{\mathsf{pf}} = \frac{\Pr(u \,|\, \langle r, t, q\rangle, \mathcal{C}_t^{\mathsf{pf}}) - \Pr(u \,|\, \langle r, t, q\rangle, \mathcal{C}_t^{\mathsf{basic}})}{\Pr(u \,|\, \langle r, t, q\rangle, \mathcal{C}_t^{\mathsf{basic}})} \tag{4.23}$$

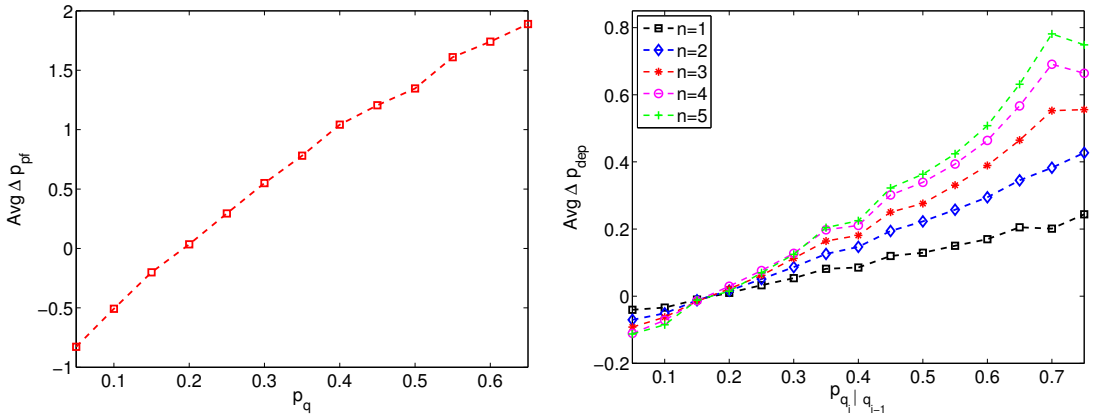
where

$$\Pr(u \,|\, \langle r, t, q\rangle, \mathcal{C}_t^{\mathsf{basic}}) = \frac{1}{|\{u \in \mathcal{U} \,|\, whereis(u, t) \in r\}|}, \tag{4.24}$$

and similarly,

$$\Delta p_{\mathsf{dep}} = \frac{\Pr(u \,|\, \langle r, t, q\rangle, \mathcal{C}_t^{\mathsf{dep}}) - \Pr(u \,|\, \langle r, t, q\rangle, \mathcal{C}_t^{\mathsf{pf}})}{\Pr(u \,|\, \langle r, t, q\rangle, \mathcal{C}_t^{\mathsf{pf}})}. \tag{4.25}$$

In Figure 4.6, we show how the correctness increase ratio changes with issuers' *a priori* preferences and the dependency between the last two queries. With respect



(a) Δp_{pf} vs. $\Pr_u(q)$. (b) Δp_{dep} vs. $\Pr(q_i \,|\, q_{i-1})$ and n.

Figure 4.6: Impact of user profiles and query dependency on Δp.

to query dependency, we also illustrate the impact of the history window sizes (see Figure 4.6(b)). The results are obtained by a simulation with $8,000$ requests. We divide the requests into clusters according to the *a priori* preference or query dependency of the issuers when sending the requests. Specifically, we set $p_q = 0.05 \cdot$ *cid* where *cid* ($1 \leq$ *cid* ≤ 20) is the identifier of a cluster to be the maximum value of issuers' *a priori* preference allowed in the cluster *cid*. For example, if $p_q = 0.15$, the issuer of any request in the cluster has an *a priori* preference between 0.1 and 0.15 with respect to the issued query. Similarly, we define $\Pr_{q_i|q_{i-1}} = 0.05 \cdot$ *cid* to represent the maximum query dependency allowed in cluster *cid* when the issuers issue the queries. Figure 4.6 depicts the average Δp_{pf} and Δp_{dep} of the generalised requests in each cluster satisfying k-anonymity with $k = 10$ and with 2.6% of the users being active.

We observe that the curves in Figure 4.6(a) and Figure 4.6(b) follow two similar patterns. First, the CIR increases monotonically when ρ grows. Second, the average correctness increase ratio reaches 0 when the *a priori* preferences and query dependency fall into the interval between 0.15 and 0.2. This is due to the fact that users' average *a priori* preference on each type of queries ($p_u(q_i)$) is around 17%. With regard to Δp_{pf}, the issuer with an *a priori* preference of 0.17 will eliminate his difference from the other users in the same region as the average of their *a priori* preferences is also close to 0.17. For Δp_{dep}, the little difference between $\Pr_u(q_i \,|\, q_{i-1})$ and $\Pr_u(q_i)$ eliminates the influence of query dependency.

We can see that Δp_{dep} is also sensitive to the size of history windows in Figure 4.6(b). Larger history windows lead to bigger correctness increase ratios when the dependency between the last two queries (i.e., $\Pr_u(q_i \mid q_{i-1})$) is bigger than 0.17. For instance, for the requests with query dependency between 0.3 and 0.4, the average value of Δp_{dep} increases by 0.051, 0.036, 0.031 when n grows from 1 to 2, from 2 to 3, from 3 to 4, respectively. By more experiments with larger n, we can show that bigger window sizes do not necessarily lead to more privacy leakage. For instance, when n is set to 5, the average increase of CIR is 0.029 which is almost the same as the case of $n = 4$.

From the above discussion, we can conclude that if a user issues a query with a large preference value or high dependency on the last queries, he will have less privacy if

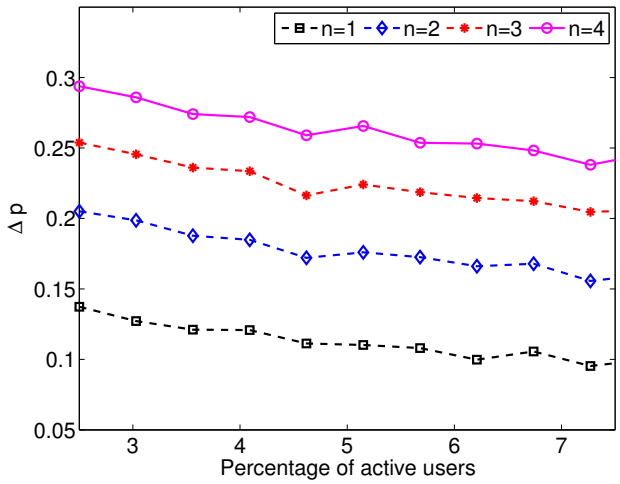


Figure 4.7: Δp vs. #active users and n.

the adversary adopts our framework. This also shows that our framework is useful to increase the likelihood of attackers to correctly learn the real issuers although we have negative CIRs when users issue queries independently from their profiles or last queries. This is because in most of the cases, users' behaviour should be consistent with their profiles and past habits.

Beside the size of history windows, the number of active users has impact on Δp_{dep} as well. It decreases when there are more active users issuing LBS requests, but the influence becomes smaller with larger history windows. Figure 4.7 shows that the average Δp_{dep} decreases by 30%, 24% 19% and 18% for $n = 1, 2, 3$ and 4, respectively, when the percentage of active users increases from 2.5% to 7.5%. This is because more active users lead to more observed requests added into users' observed request traces and mixed with users' real requests, while bigger history windows have larger chances to include users' real requests.

4.7.2 Effectiveness of the new privacy metrics

In this section we discuss the features of our privacy metrics in terms of (1) area of the generalised regions and (2) issuers' posterior probabilities. To compare the metrics presented in Section 4.5, we define a normalised value *norm*: *norm*=k for query-dependent k-ABS while *norm*=2^β for β-EBA and *norm* = $\frac{1}{\alpha}$ for α-USI. In the following experiments, we take $\mathcal{C}_t^{\mathsf{dep}}$ as the knowledge of the adversary due to its large coverage of contextual information.

Experiment setting. We set the percentage of active users to 2.6% and use the first 1,000 requests after 8,000 requests have been observed. Each number shown in the following discussion is an average of the 1,000 samples.

Recall that in the generalisation algorithm kABS (see Algorithm 4.1) we make use of a clustering algorithm to calculate the set of users with similar *a priori* probabilities. Clustering has been extensively studied in the literature and a number of clustering algorithms have been proposed to satisfy different properties, e.g., density-based and distribution-based [HK00]. In the case of generalising LBS re-

quests, the chosen clustering algorithm should satisfy at least two properties. First, the clustering algorithm should be efficient because LBS responses need to be sent back to uses in real time. Second, we need a strict partitioning clustering algorithm as each user should belong to exact one cluster.

In the implementation of kABS, we use the K-means clustering algorithm [Mac67]. This is mainly due to its linear time complexity with the number of users. Its main idea is to choose K centroids, one for each cluster. In our algorithm, the K centroids are selected randomly among the users. Then each user is associated to the nearest centroid according the difference between their *a priori* probabilities, which results in K clusters. The centroids of these K clusters are updated as the new centroids based on which all users re-calculate their centroids to associate. The process continues until the centroids remain unchanged between two consecutive iterations. In our case, K is selected and fixed by the anonymiser. In fact, it defines the 'similarity' in the definition of k-ABS in Section 4.5, i.e., ϵ. The larger K is, the smaller ϵ becomes.

In order to determine a proper value of K, we run our kABS algorithm by assigning different values to K. In Figure 4.8, we show the changes of the average distance between any two users' *a priori* probabilities in the calculated clusters and the area of the generalised regions along with K. It can be seen that a larger K enables users to have closer *a priori* probabilities but leads to larger generalised areas. In addition, the area increases faster than the decrease of the distance. Considering the relatively small generalised regions and the similarity between users in the resulted clusters, we set K to 10 in the following experiments.

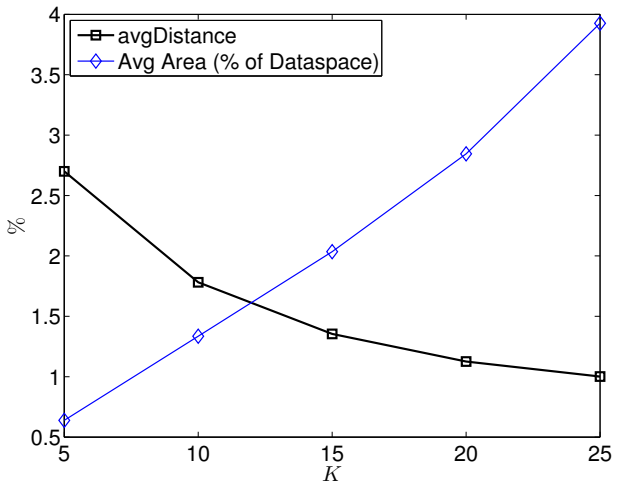
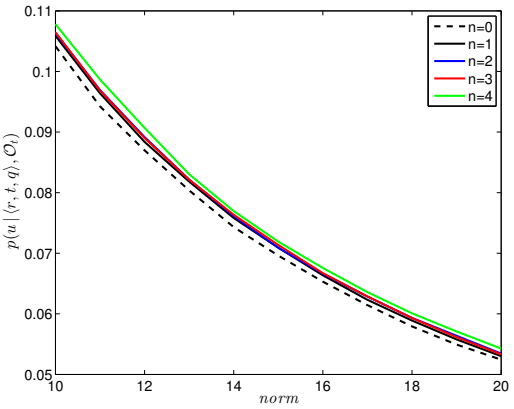


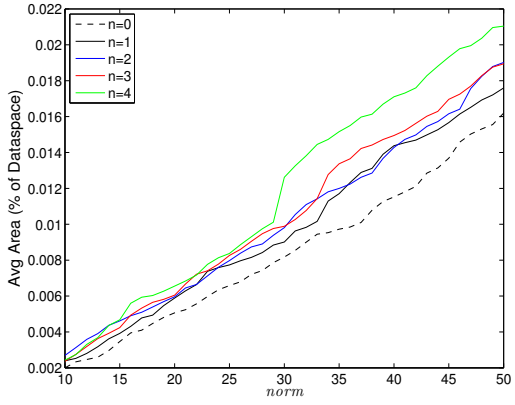
Figure 4.8: The impact of K.

Impact of history window sizes. From the above discussion, we learn that users will have less query privacy when larger history windows are used in our framework. Figure 4.9 shows how issuers' posterior probabilities and the area of generalised regions change according to the normalised value *norm* and the history window size n. Note that when $n = 0$, the generalisation algorithm only considers users' *a priori* preference.

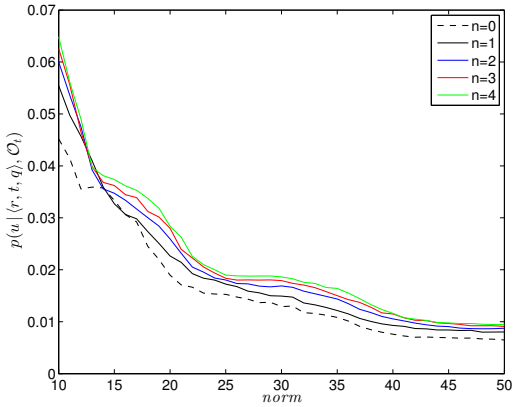
For k-ABS, issuers' posterior probabilities are about $\frac{1}{k}$ as the generalised regions have at least k users with similar posterior probabilities. However, after taking



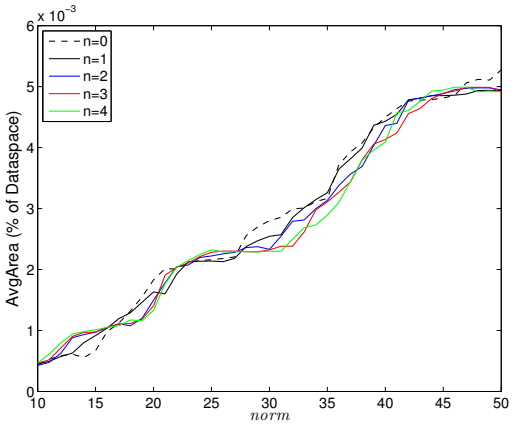
(a) $\Pr(u\,|\,\langle r,t,q\rangle,\mathcal{O}_t)$ vs. n (k-ABS).



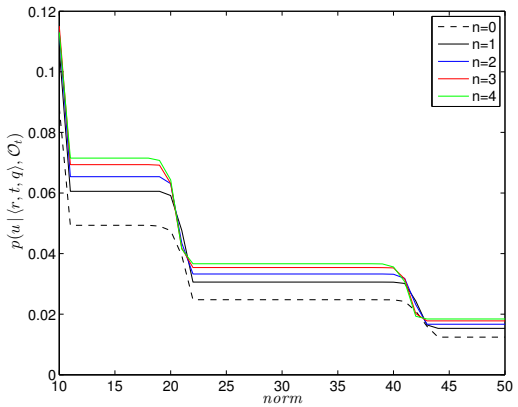
(b) Average area vs. n (k-ABS).



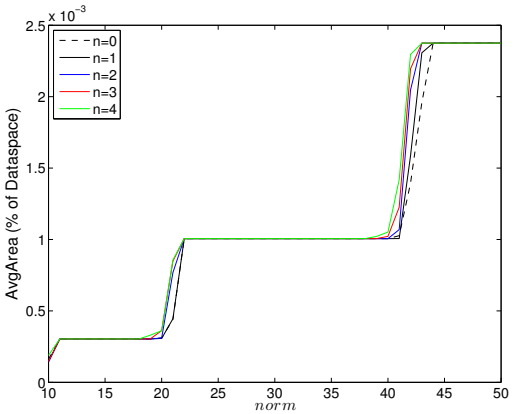
(c) $\Pr(u\,|\,\langle r,t,q\rangle,\mathcal{O}_t)$ vs. n (α-USI).



(d) Average area vs. n (α-USI).



(e) $\Pr(u\,|\,\langle r,t,q\rangle,\mathcal{O}_t)$ vs. n (β-EBA).



(f) Average area vs. n (β-EBA).

Figure 4.9: Impact of history window size n.

a closer look, we can find that a larger n leads to a larger distance to $\frac{1}{k}$. This is because larger history windows make the issuers' posterior probabilities more different from the others, which in turn makes it more difficult to find users with similar posterior probabilities. This also explains why the generalised regions become larger with larger history windows as shown in Figure 4.9(b).

For α-USI, issuers' posterior probabilities are always below $\frac{1}{norm}$, which satisfies its definition (see Figure 4.9(c)). Moreover, issuers' posterior probabilities become

larger when more historical observed requests are explored. However, the area of generalised regions differs little between different history window sizes (see Figure 4.9(d)). This is because the increase of the posterior probabilities is too small to initiate the computation of a new region.

For β-EBA, issuers' posterior probabilities can remain almost unchanged in some segments of the curves. The projection of the middle point of such a segment on axis *norm* has an logarithm of integer, such as 16 and 32 (see in Figure 4.9(e)). Similar to k-ABS, larger history windows increase the issuers' posterior probabilities, which leads to smaller entropy. This can be seen from Figure 4.9(f) where the generalised regions of larger n double their sizes earlier than the regions of smaller n.

We can also observe from Figure 4.9 that for the same value of *norm*, although the metric β-EBA cannot always ensure issuers' posterior probabilities as close to $\frac{1}{k}$ as k-ABS, the area of generalised regions is about ten times smaller than that of k-ABS and only half of that of α-USI. Since bigger regions lead to worse quality of service, this indicates that a balance between privacy protection and quality of services needs to be considered in practice.

Impact of query dependency. The protection of issuers' privacy varies with issuers' query dependency. Figure 4.10 plots posterior probabilities and average area of generalised regions for issuers with different levels of query dependency. The results are collected with the history window size n=3. Our general observation is that issuers with larger dependencies have bigger posterior probabilities and larger generalised regions.

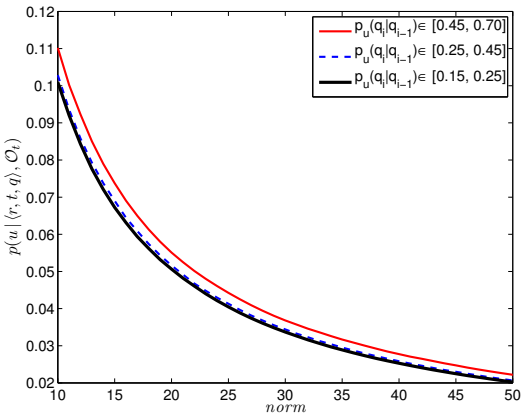
Table 4.2 summarises the corresponding average increases (in percentage) for issuers with high (≥ 0.45) and medium ($0.25 - 0.45$) dependencies, when compared with those with low dependencies (≤ 0.25). The table shows that posterior probabilities of the issuers, when β-EBA is used, are more sensitive to the degree of dependency (43.1% increase for high-level dependency), while the generalised regions are more sensitive to dependency (62.9% increase for high-level dependency) when k-ABS is used.

Table 4.2: Increases of posterior probabilities and area of generalised regions.

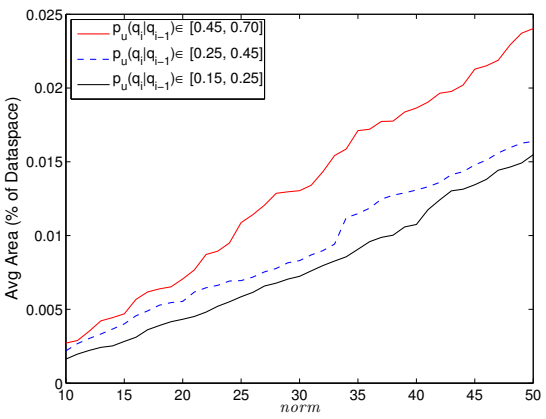
	k-ABS		β-EBA		α-USI	
	medium	high	medium	high	medium	high
Posterior Prob.	2.1%	9.5%	11.1%	43.1%	11.9%	40.0%
Avg Area	21.3%	62.9%	23.3%	30.1%	10.7%	19.1%

Performance of the proposed generalisation algorithm. In Figure 4.11, we present the performance of our generalisation algorithms to deal with users' various requirements. For the sake of comparison, we show in Figure 4.11 the performance of the algorithms when contextual information is set to $\mathcal{C}_t^{\mathsf{pf}}$ and $\mathcal{C}_t^{\mathsf{dep}}$, respectively. The computation time recorded is the average time per request based on executions with the same 100 requests.

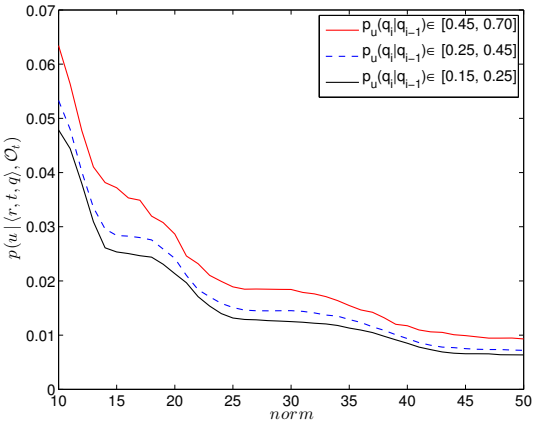
As discussed in Section 4.6, it is necessary to update the status of each user when dynamic contextual information is explored. For instance, observed request traces



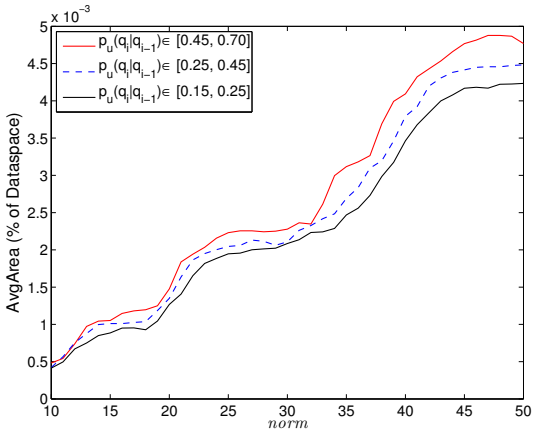
(a) $\Pr(u\,|\,\langle r,t,q\rangle,\mathcal{O}_t)$ vs. $\Pr(q_i\,|\,q_{i-1})$ (k-ABS).



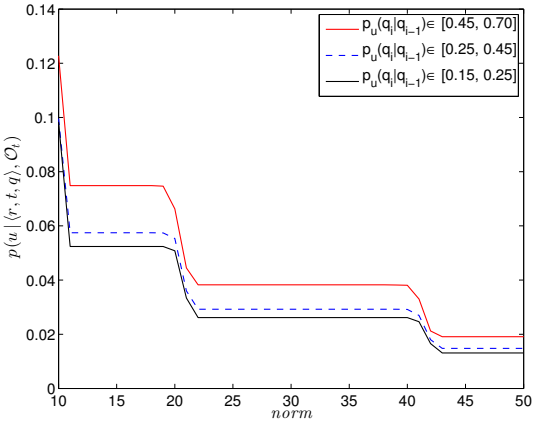
(b) Average area vs. $\Pr(q_i \mid q_{i-1})$ (k-ABS).



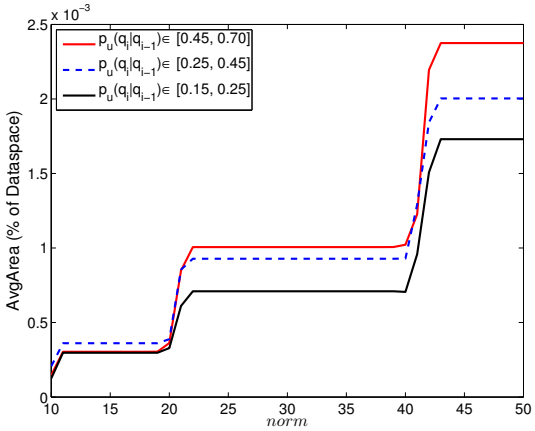
(c) $\Pr(u\,|\,\langle r,t,q\rangle,\mathcal{O}_t)$ vs. $\Pr(q_i\,|\,q_{i-1})$ (α-USI).



(d) Average area vs. $\Pr(q_i\,|\,q_{i-1})$ (α-USI).



(e) $\Pr(u\,|\,\langle r,t,q\rangle,\mathcal{O}_t)$ vs. $\Pr(q_i\,|\,q_{i-1})$ (β-EBA).



(f) Average area vs. $\Pr(q_i\,|\,q_{i-1})$ (β-EBA).

Figure 4.10: Impact of dependency $\Pr(q_i\,|\,q_{i-1})$.

and the corresponding posterior probabilities have to be updated for each request when $\mathcal{C}_t^{\mathsf{dep}}$ is used. This is time-consuming, especially when the initial region is huge and contains a large number of users. In our implementation, we reduce the computation overhead by restricting the size of initial regions. The number of users located in an initial region is fixed as ten times as many as what users require for. For instance, for k-ABS, if k=10, then we first call k-anonymity generalisation

algorithm to get an initial region with 100 users. As the generalisation algorithm is deterministic, which means for any user in a generalised region, it always returns the same region. Thus, our implementation does not have the "outlier" problem [MBFW07].

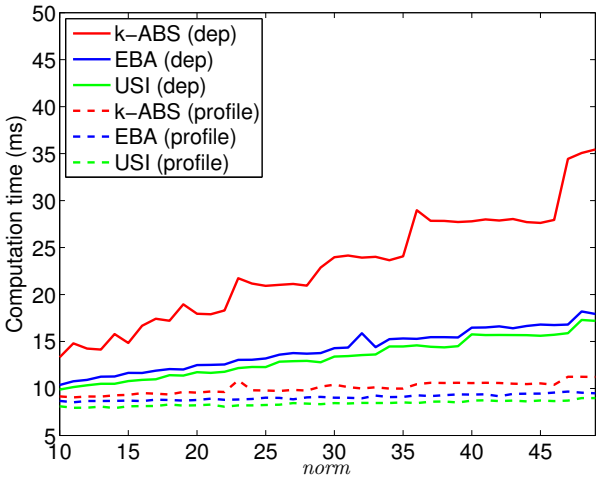


Figure 4.11: Average computational time (history window $n = 3$).

From Figure 4.11, we can see that the computation time increases as *norm* gets bigger. This is because the algorithm has to consider larger initial regions and more users are involved in the calculation of dependency-based posterior probabilities. For β-EBA and α-USI, about 20ms are needed when *norm*=50, while k-ABS requires more time (around 35ms) as the K-means clustering algorithm is executed first to find similar users. When compared to the original algorithms, the computation time increases by about two times for β-EBA and α-USI while it is about four times for k-ABS when *norm*=50.

There are some ways to improve the efficiency of our implementation. For instance, we can use better data structures to maintain users' status. We can expect that with a powerful anonymiser our algorithms are efficient enough to handle concurrent requests and give real-time responses.

4.8 Related Work

In this section, we investigate the state-of-the-art about query privacy analysis on contextual information and area generalisation algorithms.

4.8.1 Query privacy and request generalisation

Protecting users' query privacy is essentially to prevent the adversary from learning their issued queries. A number of techniques have been proposed in the literature to protect query privacy and they can be classified into three main groups: *dummy-based* [KYS05, YJHL08], *generalisation* [GG03, BLPW08, XKP09, WL09, PL11] and *cryptographic transformation* [GKK+08]. The dummy-based methods forge dummy requests with different queries such that the real queries are hidden in the

dummy ones. The idea of generalisation is to hide the real issuer in a number of users such that he is indistinguishable from the others from the view of the adversary. The concept of k-anonymity has been extensively studied in this group. In the methods exploring cryptographic transformation, users' queries are encrypted and remain secret for LBS providers so as to offer strong privacy protection. All of these methods introduce extra processing overhead. For instance, Ghinita et al. [GKK+08] build a protocol based on computational private information retrieval (cPIR) : their protocol requires one extra round of communication between users and LBS providers and imposes additional computation overheads on both sides due to the encryption of queries and decryption of responses.

The notion of k-anonymity was originally proposed by Samarati and Sweeney in the field of database privacy [Sam01]. The idea of k-anonymity is to guarantee that a database entry's identifier is indistinguishable from other $k-1$ entries. However, this method does not always work. For instance, the fact that an HIV carrier is hidden in k carriers does not protect his infection of the virus. Further research has been done to fix this problem [LLV07]. In the context of privacy in LBSs, k-anonymity was first studied by Gruteser and Grunwald [GG03]. Its purpose is to compute a region containing at least other $k-1$ users (i.e., *area generalisation*) and replace the issuer's location with it. Because of its simplicity, k-anonymity has been studied and refined in many ways. For instance, Tan et al. define information leakage to measure the amount of revealed location information in spatial cloaking, which quantifies the balance between privacy and performance [TLM09]. Xue et al. [XKP09] introduce the concept of *location diversity* to ensure generalised regions to contain at least ℓ semantic locations (e.g., schools). Many generalisation methods have been proposed to provide protection satisfying ℓ-diversity [BLPW08]. However, deeper understanding of k-anonymity reveals its drawbacks in preserving location privacy. Shokri et al. analyse the effectiveness of k-anonymity in protecting location privacy in different scenarios in terms of adversaries' background information [STD+10], i.e., *real-time location information*, *statistical information* and *no information*. They show its flaws which the adversary can exploit to infer users' current locations and conclude that spatial cloaking (e.g., k-anonymity) is only effective for protecting query privacy. As a consequence, in this work we use area generalisation *only* to protect query privacy.

4.8.2 Context-aware privacy analysis

The effectiveness of area generalisation can be compromised when the adversary has access to auxiliary contextual information. In fact, area generalisation guaranteeing k-anonymity is proposed to protect query privacy against the adversary who has users' real-time locations in their knowledge. Mascetti et al. [MBFW07] identify the 'outlier' attack on some generalisation algorithms if the adversary learns their implementations. k-anonymity is violated because the algorithms cannot ensure that all the potential issuers have the same generalised area as the real issuer. Shokri et al. [STBH11] use mobility patterns modelled as Markov chains of location transition and propose a probabilistic framework to de-anonymise generalised traces. Personal information (e.g., gender, job, salary) has been becoming more accessible on the Internet, e.g., due to online social networks such as Facebook and

LinkedIn, and can also serve as a type of contextual information which we call *user profiles*. Shin et al. [SAV08, SAV11] identify the concern of query privacy caused by user profiles and propose metrics based on k-anonymity by restricting similarity levels between users in terms of their profiles.

The contextual information (e.g., user profiles and generalisation algorithms) mentioned above is irrelevant to users' past LBS requests. Actually LBS requests can also be explored by the adversary to refine his guess on the issuers. Two types of LBS requests have been studied in the literature: *associated requests* [CZBP06, BMW+09, DRRW10a] and *recurrent requests* [RPBJ09]. Requests are associated once they are recognised as issued by a same (anonymous) user, which can be achieved for example by multi-target tracking techniques [HGXA07] or probabilistic reasoning [STBH11]. By calculating the intersection of all associated requests' anonymity sets the adversary can reduce the number of possible issuers. To handle such privacy threats, Bettini et al. [CZBP06, BMW+09] introduce *historical k-anonymity*, which is then extended for continuous LBSs by Dewri et al. [DRRW10a]. Historical k-anonymity aims to guarantee that associated requests share at least k fixed users in the generalised regions. Requests are recurrent when they are issued at the same time. When multiple recurrent requests contain the same query and region, the protection of query privacy offered by spatial cloaking (e.g., k-anonymity) will be degraded [RPBJ09]. For instance, in the extreme case, when all users in a region send an identical query, no user has query privacy. Riboni et al. [RPBJ09] identify the threat and make use of t-closeness to guarantee that the distance between the distribution over the queries from an issuer's generalised region and that of the whole region is below a threshold. Dewri et al. [DRRW10b] identify a scenario in continuous LBSs which has both associated and recurrent requests. They propose m-invariance to ensure that in addition to k fixed users shared by the associated requests, at least m different queries are generated from each generalised region.

4.8.3 Area generalisation algorithms

The first generalisation algorithm called IntervalCloaking is designed by Gruteser and Grunwald [GG03]. Their idea is to partition a region into quadrants with equal area. If the quadrant where the issuer is located contains less than k users, then the original region is returned. Otherwise, the quadrant with the issuer is taken as input for the next iteration. The algorithm CliqueCloak [GL08] is proposed by Gedik and Liu in which regions are generalised based on the users who have issued queries rather than all potential issuers. The major improvement is that this algorithm enables users to specify their personal privacy requirements by choosing different values for k. Mokbel et al. [MCA07, CMA09] design the algorithm Casper which employs a quadtree to store the two-dimensional space. The root node represents the whole area and each of other nodes represent a quadrant region of its parent node. The generalisation algorithm starts from the leaf node which contains the issuer and iteratively traverses backwards to the root until a region with more than k users is found. Another algorithm nnASR [KGMP07] simply finds the nearest k users to the issuer and returns the region containing these users as the anonymising spatial region.

The above algorithms suffer from a particular attack called "outlier problem" [Ber05], where the attackers have the generalisation algorithms and users' spatial distribution as part of their knowledge. An algorithm against this attack needs to ensure that for any user in the anonymity set it returns the same region. Kalnis et al. design the first algorithm called hilbASR that does not suffer from the outlier problem [KGMP07]. The algorithm exploits the Hilbert space filling curve to store users in a total order based on their locations. The curve is then partitioned into blocks with k users. The block with the issuer is returned as the generalised region. Mascetti et al. propose two algorithms, dichotomicPoints and grid, which are also secure against the outlier problem [MBFW07]. The former iteratively partitions the region into two blocks until less than $2k$ users are located in the region while the latter draws a grid over the two-dimensional space so that each cell contains k users and returns the cell with the issuer. Because of the simplicity of implementation and the relatively smaller area of the generalised regions, we adopt and extend these two algorithms in our algorithm design (see Section 4.6). The area of generalised regions is usually used to measure the quality of the LBS response, as smaller regions lead to more accurate results and less communication overhead.

4.9 Conclusion

In this chapter, we have developed a formal framework for query privacy analysis exploring contextual information. In the framework, we systematically categorise contextual information and propose a probabilistic way to model the adversary's attacks on query privacy. Specifically, we use a posterior probability distribution to describe the knowledge learnt by the adversary about the issuers after the analysis. This interpretation allows us to define new metrics for query privacy from different perspectives, which also facilitate users to flexibly and precisely express their privacy requirement.

We took two types of contextual information to exemplify the application of our framework. One application focuses on user profiles while the other one is further extended with contextual information: query dependency, which has not been investigated in the literature. To protect query privacy we have designed new spatial generalisation algorithms to generalise requests which can satisfy users' privacy requirements in various metrics.

Through experiments, we have shown that (1) our framework is effective to increase the correctness of the adversary's guess on real issuers; (2) the newly identified query dependency does cause privacy leakage about users' queries; (3) the proposed metrics are effective to protect users' query privacy; and (4) the generalisation algorithms are efficient.

For experiments, we made use of simulated datasets about users' movements and request traces due to the lack of real-life data with respect to LBSs. This causes some difficulties for us to test the impact of time intervals between requests. As part of our future work, we want to check whether we can collect and use users' logs in Geo-social networks in order to have a more comprehensive validation of our work.

Part III

Location Privacy

5

Activity-targeted Location Privacy Attack

In the previous chapter, we discussed the threat to query privacy by assuming a strong adversary who knows in real time the locations of users. With such an adversary, we studied the worst-case breach of query privacy when users' locations are public. An example of a public location is the home address of a user, where he usually returns at the end of the day. When users' locations are not accessible, the adversary has to learn their whereabouts before applying our methods given in Chapter 4. Meanwhile, users' whereabouts themselves help the adversary to further peek users' personal life.

This chapter is about location privacy. We study a new threat on location privacy that reveals where a users has stayed and how much time he has spent in these places. From such information, attackers can discover a user's activities, for example, visiting a doctor, doing shopping, or being at work. Compared to existing works in the literature, our attack relies only on the locations that users spontaneously send out with their LBS requests. Our attack works even if LBS requests are protected by existing privacy preserving methods.

5.1 Introduction

It has been widely recognised that the exposure of locations can threaten users' privacy [LFK05, Kru07]. To fight against such threat, several *location privacy preserving methods* (LPPM) [LOYK11, WXH+12] have been proposed in the past few years. In general, the idea of an LPPM is to break the link between users and their locations. For instance, An LPPM can *anonymise* our identities and *obfuscate* our locations by replacing them with pseudonyms and regions, respectively.

Meanwhile, breaching location privacy has also achieved interesting results [MBB13, CFP12]. These results show that in spite of the protection of existing LPPMs, it is still possible for attackers to associate users to their locations in some cases with relatively high confidence. For instance, Shokri et al. [STBH11] implement a tracking attack using user mobility profiles which can calculate the most likely locations of a user at selected time points. While existing attacks in the literature mostly target at deriving 'where users actually went', recent research requires us to revisit this objective from the view of practical attackers. Namely, what the adversary is really curious about with respect to location privacy is what users did during their movement, i.e., their *activities* [LOYK11]. For instance, receiving medical treatments reveals a user's poor health condition more precisely than just a visit to a hospital. Therefore, users are exposed to a new threat which targets at their *activity privacy*.

Normally, we need at least three elements to describe user's activities: *where* the user performed activities, *what* the activities are and *when* the user started and when the user finished them. To better represent these elements, two new concepts are introduced: *points of interest* (PoI) and *location semantics*. A PoI represents a place where a user may stay and perform activities while *location semantics* (e.g., hospital, school) correspond to PoIs and indicate the possible activities that a user can perform [XZLX10, CPX13] in a PoI. With these notions, the activities of a student at the University of Luxembourg in a day can be formulated as follows:

$$\textit{Dominican}, \mathsf{residence}, \text{0am-8am} \xrightarrow{bus} \textit{Campus Kirchberg}, \mathsf{school}, \text{9am-5pm} \xrightarrow{walking} \textit{Auchan}, \mathsf{supermarket}, \text{6pm-7pm} \xrightarrow{bus} \textit{Dominican}, \mathsf{residence}, \text{8pm- -}.$$

This trajectory says that the student left his residence located at Dominican at 8am and took a bus to Campus Kirchberg where he had courses from 9am to 5pm. Then he walked to Auchan for shopping before going home by bus.

There are many works proposed to infer users' activities based on their movements [XDZ09, HLY10, YCP+11, PSR+13, YCP+13]. Xie et al. [XDZ09] propose a method which derives users' activities by jointly considering users' visited PoIs and their duration time at each PoI by a mapping function. Huang et al. [HLY10] present a similar idea but differentiate the attractiveness of PoIs in terms of time periods of a day and their functionalities. Yan et al. [YCP+13] propose a platform to compute a user's *semantic trajectory* which contains all the required elements. A user's raw movement records are first divided into segments (also called episodes) according to the features of his movement, e.g., velocity. Then they enrich each episode where he stayed with a PoI which is subsequently annotated by a list of location semantic tags. As an episode may cover multiple PoIs, especially in urban areas, an algorithm is proposed to directly compute the semantic tags instead.

The existing works share a common assumption that users' raw movement records are accessible. In other words, users are assumed to expose their precise real-time locations with a high frequency. However, in general LBSs such as check-ins on Twitter and Foursquare, due to privacy concern, users will use LPPMs to modify their locations and identities before exposing them to outsiders. In addition, LBSs are requested sparsely in time. In this way, the adversary has no direct access to users' detailed travel history and the existing works will not work any more. In this thesis, we present the first piece of work that enables the adversary to infer users' activities from their exposed locations protected by LPPMs and thus perform effective attacks on users' activity privacy. From the literature, we notice that if a user's PoIs and his entering and exiting time of each PoI are available, the existing methods can be explored to annotate each PoI with the correct semantic tags [XDZ09, PSR+13, YCP+13]. Thus, we concentrate on calculating these two most important types of information.

In this chapter, we propose a new *tracking attack* on users' activity privacy. Through our attack, the adversary can *directly* learn a new form of trajectories based on her observation on users' sporadic exposed locations protected by LPPMs: *activity trajectories*. An activity trajectory contains not only the sequence of PoIs where a user performed activities but also his entering and exiting time at each of such PoIs.

5.2 System Model

In this section, we describe our extension of the formal framework in [STBH11] to model the components required to define our new tracking attack. The framework can be denoted as a quadruple $\langle \mathcal{U}, LPPM, ADV, \mathcal{M} \rangle$ where $\mathcal{U}$ is a set of users, $LPPM$ represents the set of deployed LPPMs, ADV is the adversary and $\mathcal{M}$ denotes privacy metrics.

Users. We consider a set of users $\mathcal{U} = \{u_1, \ldots, u_n\}$ who subscribe certain LBSs and move in an area. The area is partitioned into a finite set of regions which represent the locations with the minimum granularity, i.e., $\mathcal{R} = \{r_1, \ldots, r_n\}$. The size of cells is determined by many factors such as positioning devices (e.g., professional receivers or smart phones) and positioning systems (e.g., GSM or GPS). As users may request LBSs whenever needed, e.g., in Foursquare, we cannot exclude any time point from the possible location exposing time. Thus, different from the framework in [STBH11], we model the issuing time of an LBS request as a random variable whose value is chosen from $\mathcal{T}$, a totally ordered set with the least element 0. We use $[t, t']$ ($t, t' \in \mathcal{T}$ and $t \leq t'$) to represent a time period from time t to t' (including t and t').

A user's *trajectory* records his movements in space and time. We model it as a function mapping a time point in $\mathcal{T}$ to the user's location in $\mathcal{R}$ at that time, i.e., $\alpha_u : \mathcal{T} \to \mathcal{R}$. In the setting of LBSs, a user exposes his locations to request LBSs. Such an action is called an *exposure event* and can be denoted by a triple $\langle u, t, \alpha_u(t) \rangle$ if user u requested LBSs at time t while being at $\alpha_u(t)$. We call the time ordered sequence of exposure events of user u his *exposed trajectory* and denote it by $e_u^{[t,t']} = (\langle u, t_1, \alpha_u(t_1) \rangle, \ldots, \langle u, t_k, \alpha_u(t_k) \rangle)$ for $[t, t']$ where $t \leq t_i < t_{i+1} \leq t'$ $(1 \leq i < k)$.

We observe that in LBSs, such as nearby search and check-in posts, a user tends to issue requests from his PoIs, the places where he can perform an activity without much movements. According to this observation, we assume that users request LBSs from their PoIs and use Ψ_u to denote the set of user u's PoIs. Each $\psi \in \Psi_u$ is in fact an area of arbitrary shape and is composed of a set of adjacent regions in $\mathcal{R}$, i.e., $\Psi_u \subset 2^{\mathcal{R}}$.

Example 5.1. *Figure 5.1 depicts a user's movement in an area during a given time period. The red curve represents the user's original trajectory. The user moves through three PoIs which are identified by grey regions and labelled by ψ_1, ψ_2 and ψ_3. The user issues a request at time t_1 at PoI ψ_1 and another two request at t_2 and t_3 when he was at PoI ψ_3. The corresponding locations where the requests were issued are r_1, r_2 and r_3, respectively. As labelled on the trajectory, his exposed trajectory can be written by*

$$(\langle u, t_1, r_1 \rangle, \langle u, t_2, r_2 \rangle, \langle u, t_3, r_3 \rangle).$$

LPPMs. A user first sends his LBS requests to LPPMs where the exposure events are modified or distorted before being exposed to outsiders. In practice, LPPMs can be implemented locally on user devices or remotely on other trusted agents. We adopt the assumption of Shokri et al. [STBH11] that time is not

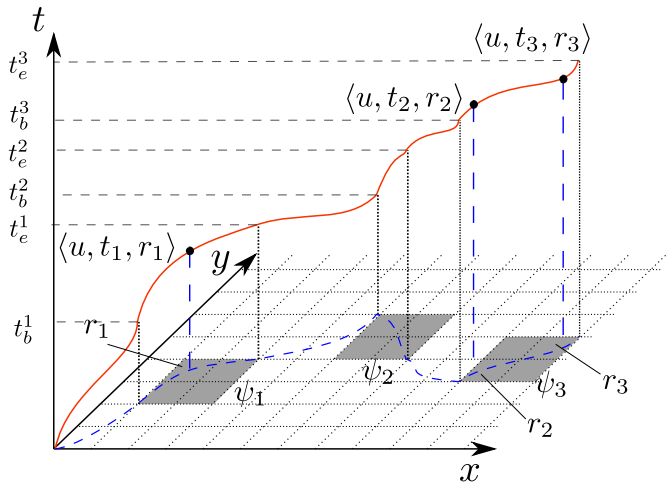


Figure 5.1: A user and his trajectory.

modified due to users' requirement for real-time responses. With our focus on popular LBSs, beside *anonymisation*, we consider two types of obfuscating LPPMs: *cloaking* and *perturbation*. Hiding requests results in frequent loss of access to LBSs and adding dummies causes extra communication overhead, which largely deteriorate the quality of LBSs.

Anonymisation replaces user identities in exposure events with pseudonyms. Although a different pseudonym can be assigned to each exposure event, as our goal is to address location privacy issues in a practical scenario, e.g., Twitter, we adopt the assumption in [STBH11] that each user is assigned a unique pseudonym. Let $\mathcal{U}' = \{u'_1, \ldots, u'_n\}$ be the set of pseudonyms. An anonymising LPPM can thus be modelled as a bijective function mapping a user identity in $\mathcal{U}$ to a pseudonym in $\mathcal{U}' = \{u'_1, \ldots, u'_n\}$, i.e., $\sigma : \mathcal{U} \to \mathcal{U}'$. This function is selected according to the uniform distribution. Thus, the probability of σ (denoted by $anony(\sigma)$) is $\frac{1}{n!}$. The obfuscating mechanism replaces the location $r \in \mathcal{R}$ in an exposure event with a *location pseudonym* $r' \in \mathcal{R}'$ according to the probability $obf(r' \,|\, r)$ where $\mathcal{R}' \subseteq 2^{\mathcal{R}}$ is the set of location pseudonyms.

User u's exposed trajectory $e_u^{[t,t']}$ is thus transformed by LPPMs to a sequence of events that can be observed by outsiders, which is called his *observed trajectory*. We denote it by $o_{u'}^{[t,t']} = (\langle u', t, r'_1 \rangle, \ldots, \langle u', t_n, r'_k \rangle)$ where $u' = \sigma(u)$ and $\forall 1 \leq i \leq k, r'_i \in \mathcal{R}'$. In the following discussion, we use $o^{[t,t']}$ to represent the set of observed trajectories of all users in $\mathcal{U}$.

The adversary. We define the adversary in terms of their *objectives*, *knowledge* and *attacks*. Intuitively, an objective of the adversary is the information she is curious about, while an attack contains the steps to achieve an objective based on her knowledge. In this thesis, we will discuss two objectives which allow the adversary to reason about users' activities more accurately. One is the linkability of each user to his pseudonym while the other is about the places a user visited and the amount of time spent in them.

Regarding the adversary's knowledge, we have the following assumptions:

(i) the adversary knows the implementation of the anonymising and obfuscating

mechanisms, i.e., *anony* and *obf*;

(ii) the adversary has access to the requests sent to LBS providers, i.e., $o^{[t,t']}$, either by collusion with LBS providers or eavesdropping;

(iii) for each user, the adversary has access to his travel history for a sufficient amount of time. This information can be obtained by side channel attacks, e.g., by breaking into the servers which users trust and expose their detailed movements to.

In the sequel, we use $\mathcal{K}$ to denote the adversary's knowledge about all the users in $\mathcal{U}$ and $\mathcal{K}_u$ to represent the knowledge about user $u \in \mathcal{U}'$.

The metric. We adopt the expected estimation error proposed by Shokri et al. [STBH11] to measure the effectiveness of attacks or users' privacy guaranteed by LPPMs. After an attack, the adversary will learn a set of possible values $\mathcal{X}$ about her objective and a probability distribution $\Pr(x \mid o)$ $(x \in \mathcal{X})$. Suppose x' is the real value of the objective. Let $\| x, x' \|$ be the distance between x and x'. Its definition depends on the type of x and x'. For pseudonyms, if the pseudonym of user u' is u', then $\| u', u'' \|$ is 0 when u'' is equal to u' and 1, otherwise. The adversary's estimation error is calculated as follows:

$$privacy(o, x) = \sum_{x \in \mathcal{X}} \Pr(x \mid o) \cdot \| x, x' \| \tag{5.1}$$

In Table 5.1 we summarise the important notations that have been introduced and those which will be given in the rest of this chapter.

5.3 Profiling Users

In this section, we propose a new model for user mobility profiles and request issuing patterns. First, our new user profiles can equip attackers with knowledge to infer users' activities more accurately. Second, compared to the discrete-time models, our model can enable us to describe in a more natural manner users' LBS requesting time which spans over the continuous time space.

5.3.1 Mobility profiles

A user's mobility profile captures his movement patterns. The idea of our new model is inspired by an intuitive observation. Namely, a user always moves with certain purposes which actually determine the places the user will visit. Furthermore, to accomplish a purpose, a user usually stays in a PoI. Therefore, a user's trajectory can be divided into two types of segments: *stay at a PoI* and *transition between PoIs*. Furthermore, this division allows us to make the following assumptions.

1. A user determines his next destination based on his past locations. In other words, the sequence of visited PoIs follows a Markov chain whose order is decided by the user's behaviour and may differ from other users.

Table 5.1: The important notations.

$\mathcal{U}$	set of users
$\alpha_u^{[t_1,t_2]}$	the original trajectory of user u in time period $[t_1, t_2]$
Ψ_u	set of PoIs of user u
$\mathcal{U}'$	set of pseudonyms
σ	anonymising strategy
$\mathcal{R}'$	set of obfuscated regions
$\mathcal{K}$	knowledge of the adversary
$\mathcal{K}_u$	knowledge of the adversary about user u
$obf(r' \mid r)$	probability that r is obfuscated as r'
$o^{[t,t']}$	set of observed trajectories of users in $\mathcal{U}$ in time period $[t, t']$
$o_{u'}^{[t,t']}$	observed trajectory with the pseudonym u' in time period $[t, t']$
$\mathcal{P}_u$	mobility profile of user u
$\delta_{min}^u(\psi, \psi')$	minimum transition time from ψ to ψ'
Ω_u	transition matrix between PoIs of user u
$\Gamma_s^u(\psi, s)$	probability density of u staying at ψ for time s
$\Gamma_t^u(\psi, \psi', s)$	probability density of u transiting from ψ to ψ' with s
$LPPM$	location privacy preserving method
ADV	the adversary
$\pi_u(\psi)$	the probability that user u is in PoI ψ
$\tau_u(\psi, t)$	probability density function of the time when u enters ψ
$a_u^{[t,t']}$	activity trajectory of user u in time period $[t, t']$
t_b^i	entering time of the ith PoI in the given PoI sequence
t_e^i	exiting time of the ith PoI in the given PoI sequence
$\mathcal{T}_b^{i,s,O}$	interval of the entering time of the ith PoI in the PoI sequence s given observed trajectory O
$\mathcal{T}_e^{i,s,O}$	interval of the exiting time of the ith PoI in the PoI sequence s given observed trajectory O
$\lambda_u(\psi)$	average number of requests from u in ψ in a time unit
$N(s)$	the length of the sequence of PoIs s
ψ_i^s	ith PoI in the sequence of PoIs s
$\Xi^{s,O}$	decomposition of observed trajectory O with respect to the sequence of PoIs s

2. We follow the common assumption in the literature about location semantic annotation: the activities which users perform in a PoI are decided by the PoI itself. Thus, since stay time depends on activities, it is determined by the current PoI.

3. The transition time between two PoIs is only determined by the source and destination. This is reasonable because transition time is mainly determined by the distance between two PoIs and factors affecting movement, e.g., traffic and weather.

4. Users require a minimum time to move between two PoIs which is restricted by the distance and available means of transport.

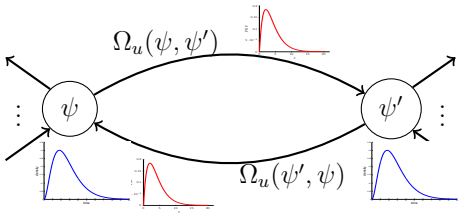


Figure 5.2: An example of $\mathcal{P}_u$.

Now based on the above discussion, we can proceed to define our model for user mobility profiles as follows:

Definition 5.1 (Mobility profile)**.** *A user u's mobility profile is represented by a tuple $\mathcal{P}_u = \langle \Psi_u, \Omega_u, \Gamma_s^u, \Gamma_t^u, \delta_{min}^u \rangle$ where*

- Ψ_u*: a finite set of points of interest;*
- $\Omega_u : \Psi_u^d \times \Psi_u \to [0, 1]$*: For any $\psi_1, \ldots, \psi_d \in \Psi_u$, $\Omega_u(\psi_1, \ldots, \psi_d, \cdot)$ is the probability of user u's next PoI after having sequentially visited $\psi_1, \ldots, \psi_d$. It holds that $\sum_{\psi \in \Psi_u} \Omega_u(\psi_1, \ldots, \psi_d, \psi) = 1$.*
- $\Gamma_s^u : \Psi_u \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$*: for any $\psi \in \Psi_u$, $\Gamma_s^u(\psi, \cdot)$ is the probability density function of the amount of time u stays at PoI ψ. It holds that $\int_{\delta \geq 0} \Gamma_s^u(\psi, \delta) d\delta = 1$.*

- $\Gamma_t^u : \Psi_u \times \Psi_u \times \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$*: for any $\psi, \psi' \in \Psi_u$, $\Gamma_t^u(\psi, \psi', \cdot)$ is the probability density function of the amount of time user u spends on transiting from ψ to ψ'. It holds that $\int_{\delta > 0} \Gamma_t^u(\psi, \psi', \delta) d\delta = 1$.*
- $\delta_{min}^u : \Psi_u \times \Psi_u \to \mathbb{R}_{>0}$*: the minimum amount of time required to transit between any two PoIs. For any $\psi \in \Psi_u$ it holds that $\delta_{min}^u(\psi, \psi) > 0$. This is interpreted as the minimum time required by user u to leave PoI ψ and to return to it.*

Figure 5.2 depicts part of user u's mobility profile $\mathcal{P}_u$ with $d = 1$ in terms of two PoIs. Intuitively, after user u enters a PoI ψ, he stays at ψ for a certain amount of time δ_s according to $\Gamma_s^u(\psi, \delta_s)$. Then, user u selects his next PoI ψ' with probability $\Omega_u(\psi, \psi')$ and the transition time δ_s from ψ to ψ' follows the density function $\Gamma_t^u(\psi, \psi', \delta_t)$. Similar to [STBH11], our mobility profiles do not consider the fact that users may behave distinctively in various time periods. For instance, an accountant works in the shopping mall where he usually does shopping on weekends. Obviously, he will stay much longer on weekdays than on weekends because the tasks of the visits are different. This can be solved by constructing individual mobility profiles for each time period.

The starting d PoIs are determined by the frequency of user u visiting the sequence. We use $\pi_u(\psi_1, \ldots, \psi_d)$ to denote the probability that user u initiates a trajectory with $(\psi_1, \ldots, \psi_d)$. There may exist other factors affecting the starting PoIs, e.g., time. Due to the continuous time space, we use $\tau_u : \Psi^d \times \mathcal{T} \to \mathbb{R}_{\geq 0}$ to denote the *probability density function* of the time when user u starts the given sequence of PoIs.

Suppose user u enters PoI ψ_1 at time point t, sequentially visits PoIs $\psi_2, \ldots, \psi_k$ and leaves ψ_k at t'. Then these movements can be represented as a sequence

$$a_u^{[t,t']} = (\langle \psi_1, t_b^1, t_e^1 \rangle, \langle \psi_2, t_b^2, t_e^2 \rangle, \ldots, \langle \psi_k, t_b^k, t_e^k \rangle),$$

where t_b^i and t_e^i are respectively the entering and exiting time points of PoI ψ_i $(1 \leq i \leq k)$. Notice that $t_b^1 = t$ and $t_e^k = t'$. Moreover, $a_u^{[t,t']}$ also satisfies three other properties:

(i) for all $d \leq i \leq k$, $\Omega_u(\psi_{i-d+1}, \ldots, \psi_i, \psi_{i+1}) > 0$ indicating that all consecutive transitions between PoIs are possible according to the mobility profile of user u;
(ii) for all $1 \leq j \leq k$, $t_e^j - t_b^j$ follows the distribution $\Gamma_s^u(\psi_j, t_e^j - t_b^j)$; and
(iii) for all $1 < j \leq k$, $t_b^j - t_e^{j-1} \geq \delta_{min}^u(\psi_{j-1}, \psi_j)$ and it is distributed in accordance with $\Gamma_t^u(\psi_j, \psi_{j+1}, t_b^j - t_e^{j-1})$.

The sequence $a_u^{[t,t']}$ can help the adversary perform a more accurate inference on u's activities: (i) it contains the sequence of PoIs whose semantics indicate users' possible activities; (ii) it involves user u's entering and exiting time of each PoI which provide more information to determine the real semantic a user used during his visit to a PoI. For this reason, we refer to $a_u^{[t,t']}$ as user u's *activity trajectory*.

For the sake of providing a concise presentation, we use the first-order Markov chain to model a user's transition between PoIs in the rest of this section as well as in the description of our new tracking attack in Section 5.4. In our validation (Section 5.6), we do not have this restriction and we calculate the order that best captures a user's mobility. Given user u's mobility profile in our new model, we can calculate the probability density function of $a_u^{[t,t']} = (\langle \psi_1, t_b^1, t_e^1 \rangle, \langle \psi_2, t_b^2, t_e^2 \rangle, \ldots, \langle \psi_k, t_b^k, t_e^k \rangle)$ as follows:

$$\begin{aligned} f_{actTraj}(a_u^{[t,t']} \,|\, \mathcal{P}_u) = & \underbrace{\pi_u(\psi_1)\tau_u(\psi_1, t_b^1)}_{\textbf{Part I}} \cdot \underbrace{\Big(\prod_{1 \leq i \leq k} \Gamma_s^u(\psi_i, t_e^i - t_b^i) \Big)}_{\textbf{Part II}} \\ & \underbrace{\cdot \Big(\prod_{2 \leq i \leq k} \Gamma_t^u(\psi_{i-1}, \psi_i, t_b^i - t_e^{i-1}) \cdot \Omega_u(\psi_{i-1}, \psi_i) \Big)}_{\textbf{Part III}} . \end{aligned} \tag{5.2}$$

Part I of Equation 5.2 is the probability density that user u starts an activity trajectory at ψ_1 at time t_b^1 while Part II specifies the joint probability density that user u stays at any PoI ψ_i for $t_e^i - t_b^i$. Part III expresses user u' transitions between PoIs, i.e., the joint probability density that u sequentially travels $\psi_2, \ldots, \psi_k$ with the given transition time.

5.3.2 Request issuing patterns

Users have some patterns with respect to when and where they prefer to request LBSs. For instance, the local-search services of nearby restaurants are usually requested during lunch or dinner time from residential areas. There are three main factors deciding whether to expose a location: the location itself, the time

period, and the type of queries. In the extended framework of [STBH11] to sporadic LBSs [STD+11], a binary probability distribution is assigned to each time point and governs whether a request is issued at that time point. This model is reasonable as the considered time space is discrete and finite. In this thesis, we relax the assumption and allow a user to request LBSs at any time during his stay in a PoI. We assume that the process of issuing requests is controlled by some probability distribution. As the characteristics of the process, e.g., the number of requests issued in a time unit, may differ between users and PoIs, distributions for different combinations of users and PoIs should be used.

We present a possible model profiling users' request issuing patterns in this section. Notice that if extra knowledge on the distributions is available, it can be easily incorporated into our user profiles. To have a concise presentation, we only consider locations as they are the most influential factor for LBSs such as check-ins in places of interest. With this assumption we model the number of requests issued in a time unit (e.g., an hour) by user u in PoI ψ as a Poisson process with rate $\lambda_u(\psi)$. As a result, the amount of time between requests is exponentially distributed with $\lambda_u(\psi)$ which can be interpreted as the average number of requests from u in a time unit during his stay in ψ.

Suppose that user u enters PoI ψ at time t_b and exits it at t_e. Furthermore, let u' be the pseudonym assigned to the user according to the anonymising strategy σ selected by the deployed anonymising LPPM. Then we can calculate the probability density function that user u, anonymised as u', issues a sequence of observed events $obsSeq = (\langle u', t_1, r'_1\rangle, \ldots, \langle u', t_k, r'_k\rangle)$ within time period $[t_b, t_e]$ during his stay at PoI ψ as follows:

$$\begin{aligned} & f_{issue}(obsSeq \,|\, [t_b, t_e], \psi, \sigma(u) = u') \\ = & \underbrace{\lambda_u(\psi)\mathrm{e}^{-\lambda_u(\psi)(t_1 - t)} \cdot \prod_{i=2}^{k} \lambda_u(\psi)\mathrm{e}^{-\lambda_u(\psi)(t_i - t_{i-1})}}_{\textbf{Part I}} \cdot \underbrace{\mathrm{e}^{-\lambda_u(\psi)\cdot(t' - t_k)}}_{\textbf{Part II}} \cdot \\ & \underbrace{\prod_{\langle u', t_i, r'_i\rangle \in obsSeq} \sum_{r \in \psi} \Pr(r \,|\, \psi) \cdot obf(r, r')}_{\textbf{Part III}}, \end{aligned} \tag{5.3}$$

The above equation has three parts. In Part I and Part II the requests are issued in accordance with the probabilistic model described above, i.e., the issuing time of subsequent requests are independently and exponentially distributed with $\lambda_u(\psi)$. Part I is the probability density of user u issuing k requests within the time period $[t_b, t_e]$ at $t_1, \ldots, t_k$. Part II gives the probability that no requests are issued in the remaining time period $[t_k, t_e]$. Parts I and II together provide the probability density of user u issuing exactly k requests within $[t_b, t_e]$ at $t_1, \ldots, t_k$. Part III is the joint probability that the location pseudonyms r'_i $(1 \leq i \leq k)$ are output by the deployed obfuscating LPPM. $\Pr(r \,|\, \psi)$ is the probability that user u is located at r given that he is now in PoI ψ. If no further information is available, we assume that a uniform distribution on all r in ψ, i.e., $\Pr(r \,|\, \psi) = \frac{1}{|\psi|}$. Note that r is a region with the minimum granularity and ψ is a set of such regions.

Example 5.2. *In our running Example 5.1, user u issued one request at r_1 in PoI ψ_1 at time t_1. Suppose r_1 is obfuscated to r'_1. Then from the view of the adversary,*

the probability density of user u issuing this observed request during the stay in ψ_1 can be calculated as follows:

$$\begin{aligned} & f_{issue}((\langle u', t_1, r_1' \rangle) \,|\, [t_b^1, t_e^1], \psi_1, \sigma(u) = u') \\ =& \lambda_u(\psi_1) \cdot e^{-\lambda_u(\psi_1)(t_1 - t_b^1)} \cdot e^{-\lambda_u(\psi_1) \cdot (t_e^1 - t_1)} \cdot \sum_{r \in \psi_1} Pr(r \,|\, \psi_1) \cdot obf(r_1' \,|\, r). \end{aligned}$$

5.4 A New Tracking Attack

In this section, we propose a new tracking attack making use of our new model for user profiles. Intuitively, the objective of this attack is to find the most likely activity trajectories for all users. Our attack allows the adversary to learn not only the PoIs where users perform activities but also directly the entering time and exiting time of such PoIs. In the following discussion, we suppose that the adversary has learnt the observed trajectories of all users in $\mathcal{U}$ in the time period $[t, t']$, i.e., $o^{[t,t']}$. For the purpose to have a concise presentation, we assume that all users enter a PoI at time t and exit a PoI at t'. This assumption captures users' daily life that they usually start and finish work at almost the same time. Note that our attack can be extended to cover more general cases. In the rest of this section, we omit $[t, t']$ in the notations when it is clear from the context.

We split the tracking attack into two sequential steps following the same reasoning as in [STBH11, STD+11]: first *de-anonymisation* and then *de-obfuscation*. De-anonymisation is to find the most likely anonymising strategy mapping user identities to pseudonyms while de-obfuscation aims to find the most probable activity trajectories given users' pseudonyms.

5.4.1 De-anonymisation

The goal of the adversary is to find the most probable mapping function σ^* from $\mathcal{U}$ to $\mathcal{U}'$ given users' observed trajectories o and the profiles of the users in $\mathcal{U}$. This can be formulated as the following optimisation problem:

$$\sigma^* = \arg\max_{\sigma} \ \Pr(\sigma \,|\, o, \mathcal{K}). \tag{5.4}$$

Although the main idea of de-anonymisation is similar to that in [STBH11], we need to propose a new and efficient calculation process due to our consideration of continuous time space and new model for user profiles. We use $F_{obv}(o \,|\, \sigma, \mathcal{K})$ to represent the probability density of o given the anonymising strategy σ and the adversary's knowledge $\mathcal{K}$. By applying the Bayesian theorem, we have that

$$\Pr(\sigma \,|\, o, \mathcal{K}) \propto F_{obv}(o \,|\, \sigma, \mathcal{K}) \cdot \Pr(\sigma), \tag{5.5}$$

where the proportionality factor is independent of σ and can be considered constant. Due to the assumption that the choice of the anonymising strategy follows a uniform distribution, we have that $\Pr(\sigma)$ is $\frac{1}{n!}$. Thus, the optimisation is reduced to finding σ that maximises $F_{obv}(o \,|\, \sigma, \mathcal{K})$. Since users are independent of each other when travelling and exposure events are anonymised and obfuscated

independently, $F_{obv}(o\,|\,\sigma, \mathcal{K})$ can be factorised into a product of probability density functions as follows:

$$F_{obv}(o\,|\,\sigma, \mathcal{K}) = \prod_{u' \in \mathcal{U}'} f_{obv}(o_{u'}\,|\,\mathcal{K}_u, [t, t'], \sigma(u) = u'). \tag{5.6}$$

Note that $f_{obv}(o_{u'} \mid \mathcal{K}_u, [t, t'], \sigma(u) = u')$ is the probability density that user u with pseudonym u' issues the observed trajectory $o_{u'}$ in $[t, t']$. In this way, the problem can be further reduced to calculating the mapping function σ^* which maximises the above product. In fact, de-anonymisation can be formulated as assigning pseudonyms in $\mathcal{U}'$ to users in $\mathcal{U}$. If we take $-\log f_{obv}(o_{u'}\,|\,\mathcal{K}_u, [t, t'], \sigma(u) = u')$ to be the cost of assigning u' to u, then the optimisation can be seen as a *minimum weight assignment problem* and the existing solutions can be exploited. Specifically, this is because $\arg\max_\sigma\ F_{obv}(o\,|\,\sigma, \mathcal{K})$ is equivalent to $\arg\min_\sigma\ -\log F_{obv}(o\,|\,\sigma, \mathcal{K})$.

In the following, we present an efficient method to calculate users' patterns with respect to their observed trajectories, i.e., $f_{obv}(o_{u'}\,|\,\mathcal{K}_u, [t, t'], \sigma(u) = u')$.

Calculating observed trajectory patterns. This probability density can be obtained by marginalising the joint probability density function of a user to issue an observed trajectory $o_{u'}$ and meanwhile travel an activity trajectory a_u, i.e., $f_{actTraj,obv}(a_u, o_{u'}\,|\,\mathcal{K}_u, [t, t'], \sigma(u) = u')$, over all activity trajectories. Thus, we proceed to derive the density function $f_{actTraj,obv}$ and show how it can be marginalised to obtain f_{obv}.

We notice that $f_{actTraj,obv}$ is zero for all activity trajectories that are not compatible with $o_{u'}$, where the compatibility is understood as follows. We say that an activity trajectory a_u is *compatible* with $o_{u'}$ if and only if for the time point given by any observed event in $o_{u'}$, user u is at a PoI and not in transition between two PoIs. To make further computation efficient, we introduce a scheme to consider only the activity trajectories that are compatible with $o_{u'}$.

Let $\mathcal{S}_u$ be the set of all sequences of PoIs that user u could potentially visit in the time period $[t, t']$ which are allowed by the minimum time required to move between two consecutive PoIs. If we use $N(s)$ to denote the length of sequence s and ψ_i^s be the ith PoI in s, then

$$\mathcal{S}_u = \big\{ s\,|\,\forall_{i=1,\ldots,N(s)}\, \psi_i^s \in \Psi_u,\ \sum_{i=1}^{N(s)-1} \delta_{min}^u(\psi_i^s, \psi_{i+1}^s) \leq t' - t \big\}. \tag{5.7}$$

Two remarks are in place. First, for each user $u \in \mathcal{U}$ and each time period $[t, t']$, the set $\mathcal{S}_u$ is finite since user u has finite PoIs and the minimum transition time between any two PoIs is non-zero. Second, by the definition of activity trajectories, the sequence of PoIs visited by u within any activity trajectory is contained in $\mathcal{S}_u$.

Assume that user u is assigned pseudonym u'. We proceed to consider how an observed trajectory $O \equiv o_{u'}$ could be obtained given that the user visited a sequence of PoIs $s \in \mathcal{S}_u$. The following restrictions are in place: (i) each observed event in O is issued from some PoI in s; (ii) any two consecutive events are issued either from the same PoI or the second event is issued from some subsequent PoI in s; (iii) s may contain PoIs where no events are issued.

With these restrictions, we can decompose O into $N(s)$ disjoint blocks of contiguous observed events in O. The ith block is the sequence of all observed events issued from PoI ψ_i^s. We use $\Xi_i^{s,O}$ to denote the ith block and $\Xi^{s,O} = (\Xi_1^{s,O}, \Xi_2^{s,O}, \ldots, \Xi_{N(s)}^{s,O})$ is a *decomposition* of O with respect to s. Note that for O and s, there are usually a number of different decompositions with respect to s which is actually exponential in the number of observed events in O.

We say that an activity trajectory a_u *complies* with PoI sequence s and $\Xi^{s,O}$ if the ith PoI in a_u is equal to ψ_i^s and user u enters it before issuing the first request in $\Xi_i^{s,O}$ and exits it after having issued the last request in $\Xi_i^{s,O}$. Let t_b^i and t_e^i are the entering and exiting time points of the ith PoI in a_u, then

$$t_b^i \leq \min\{t'' \,|\, \exists \langle u', t'', r' \rangle \in \Xi_i^{s,O})\}; \tag{5.8}$$

$$t_e^i \geq \max\{t'' \,|\, \exists \langle u', t'', r' \rangle \in \Xi_i^{s,O})\}. \tag{5.9}$$

These two conditions lead to a small interval for the entering (exiting) time of each PoI in s. We use $\mathcal{T}_b^{i,s,O}$ and $\mathcal{T}_e^{i,s,O}$ $(1 \leq i \leq N(s))$ to denote such time intervals for the entering and exiting time of PoI ψ_i^s, respectively. Any activity trajectory that complies with $\Xi^{s,O}$ can be obtained by assigning each time variable (i.e., t_e^i and t_b^i) a value from the corresponding time interval (i.e., $\mathcal{T}_e^{i,s,O}$ and $\mathcal{T}_b^{i,s,O}$, respectively).

We denote the lower (upper) bound of a time interval $\mathcal{T}'$ as $\ell owbnd(\mathcal{T}')$ ($upbnd(\mathcal{T}')$). Let $nextNEB(\Xi^{s,O}, i)$ $(1 \leq i < k)$ be the index of the first non-empty block in $\Xi^{s,O}$ with an index larger than i, i.e., $\min\{j > i \,|\, \Xi_j^{s,O} \neq \emptyset\}$. If no such block exists, i.e., $\Xi_j^{s,O} = \emptyset$ for all $i < j \leq k$, then we set $nextNEB(\Xi^{s,O}, i) = k$. The main idea of determining the bounds is to exploit another two principles. First, the exiting time of a PoI is larger than the entering time; second, the definition of activity trajectories requires that the time interval between two successive visited PoIs is larger than the minimum allowed transition time. Based on these two principles, we can calculate the bounds of the entering time of each PoI in s as the following:

$$i = 1,\ \ell owbnd(\mathcal{T}_b^{i,s,O}) = upbnd(\mathcal{T}_b^{i,s,O}) = t;$$

$$1 < i \leq N(s),\ \ell owbnd(\mathcal{T}_b^{i,s,O}) = t_e^{i-1} + \delta_{min}^u(\psi_{i-1}^s, \psi_i^s);$$

$$upbnd(\mathcal{T}_b^{i,s,O}) = \begin{cases} \min\{t'' \,|\, \exists \langle u', t'', r' \rangle \in \Xi_i^{s,O}\} & \Xi_i^{s,O} \neq \emptyset \\ \min\{t'' \,|\, \exists \langle u', t'', r' \rangle \in \Xi_{nextNEB(\Xi^{s,O},i)}^{s,O} \vee \\ \ t'' = t'\} - \sum_{j=i+1}^{nextNEB(\Xi^{s,O},i)} \delta_{min}^u(\psi_{j-1}^s, \psi_j^s) & \Xi_i^{s,O} = \emptyset \end{cases}$$

For the exiting time of each PoI, we can perform the following calculation:

$$i = N(s),\ \ell owbnd(\mathcal{T}_e^{i,s,O}) = upbnd(\mathcal{T}_e^{i,s,O}) = t';$$

$$1 \leq i < N(s),\ \ell owbnd(\mathcal{T}_e^{i,s,O}) = \begin{cases} \max\{t'' \,|\, \exists \langle u', t'', r' \rangle \in \Xi_i^{s,O}\} & \Xi_i^{s,O} \neq \emptyset \\ t_b^i & \Xi_i^{s,O} = \emptyset. \end{cases}$$

$$upbnd(\mathcal{T}_e^{i,s,O}) = \min\{t'' \,|\, \exists \langle u', t'', r' \rangle \in \Xi_{nextNEB(\Xi^{s,O},i)}^{s,O} \vee t'' = t'\} - \sum_{j=i+1}^{nextNEB(\Xi^{s,O},i)} \delta_{min}^u(\psi_{j-1}^s, \psi_j^s).$$

For any $1 \leq i \leq N(s)$, it holds that

$$\begin{aligned} \ell owbnd(\mathcal{T}_b^{i,s,O}) \leq & upbnd(\mathcal{T}_b^{i,s,O}) \leq \ell owbnd(\mathcal{T}_e^{i,s,O}) \\ & \wedge\, upbnd(\mathcal{T}_e^{i,s,O}) \leq \ell owbnd(\mathcal{T}_b^{i+1,s,O}). \end{aligned}$$

However, it is possible that for a non-empty block $\Xi_i^{s,O}$ in $\Xi^{s,O}$, $upbnd(\mathcal{T}_e^{i,s,O}) < \ell owbnd(\mathcal{T}_e^{i,s,O})$. In this case, there is no activity trajectory that complies with the decomposition $\Xi^{s,O}$. The corresponding probability density is thus directly set to zero during the marginalisation over all possible decompositions.

Let $a_u = (\langle \psi_1, t_b^1, t_e^1 \rangle, \ldots, \langle \psi_k, t_b^k, t_e^k \rangle)$ be an activity trajectory that complies with $\Xi^{s,O}$ and $s = (\psi_1, \ldots, \psi_k)$. In Equation 5.10, we calculate the joint probability density of user u travelling a_u (Part I) and generating the observed trajectory $O \equiv o_{u'}$ (Part II):

$$\begin{aligned} & f_{actTraj,obv}(a_u, O \,|\, \mathcal{K}_u, [t,t'], \sigma(u) = u') \\ = & \underbrace{f_{actTraj}(a_u \,|\, \mathcal{K}_u)}_{\textbf{Part I}} \cdot \underbrace{\prod_{i=1}^{N(s)} f_{issue}(\Xi_i^{s,O} \,|\, [t_b^i, t_e^i], \psi_i^s, \sigma(u) = u')}_{\textbf{Part II}}. \end{aligned} \tag{5.10}$$

With the above density function, in the following, we present a method to marginalise over all activity trajectories and obtain our target f_{obv}, i.e., the density function of user u issuing an observed trajectory.

We start with constructing the activity trajectories compatible with an observed trajectory $O \equiv o_{u'}$. We observe that the following two inference rules hold:

(i) if a_u and O are compatible, there exists s and $\Xi^{s,O}$ that a_u complies with;

(ii) if a_u *complies* with s and $\Xi^{s,O}$, then a_u is compatible with O.

These two rules allow us to construct all activity trajectories that are compatible with the observed trajectory by considering (i) each $s \in \mathcal{S}_u$; (ii) each possible decomposition of the observed trajectory with respect to s; and (iii) every possible combination of the entering and exiting time points within the time intervals determined by s and the decomposition. Therefore, we can write

$$\begin{aligned} & f_{obv}(O \equiv o_{u'} \,|\, \mathcal{K}_u, [t,t'], \sigma(u) = u') \\ = & \sum_{s \in \mathcal{S}_u} \sum_{\Xi^{s,O}} \int_{\mathcal{T}_b^{1,s,O} \times \mathcal{T}_e^{1,s,O} \times \ldots \times \mathcal{T}_b^{N(s),s,O} \times \mathcal{T}_e^{N(s),s,O}} \\ & \qquad f_{actTraj,obv}(a_u, o_{u'} \,|\, \mathcal{K}_u, [t,t'], \sigma(u) = u') \; \mathrm{d}t_e^{N(s)} \; \mathrm{d}t_b^{N(s)} \; \ldots \mathrm{d}t_e^1 \; \mathrm{d}t_b^1 \,, \end{aligned} \tag{5.11}$$

where $a_u \equiv (\langle \psi_1^s, t_b^1, t_e^1 \rangle, \ldots, \langle \psi_{N(s)}^s, t_b^{N(s)}, t_e^{N(s)} \rangle)$.

5.4.2 De-obfuscation

We present a method that the adversary may implement to find the most likely activity trajectory given a user's observed trajectory. Let σ^* be the anonymising strategy obtained by applying the de-anonymisation attack. Furthermore, let

$f_{act}(a_u \,|\, o_{\sigma^*(u)}, \mathcal{K}_u)$ be the probability density function that user u travels the activity trajectory a_u given his observed trajectory $o_{\sigma^*(u)}$ in the time period $[t, t']$. Then, the objective of this attack can be formulated as

$$\arg\max_{a_u} \; f_{act}(a_u \,|\, o_{\sigma^*(u)}, \mathcal{K}_u).$$

Subsequently, this can be rewritten as

$$\arg\max_{a_u} \; f_{act}(a_u \,|\, o_{\sigma^*(u)}, \mathcal{K}_u) = \arg\max_{a_u} \; \frac{f_{actTraj,obv}(a_u, o_{u'} \,|\, \mathcal{K}_u, [t, t'], \sigma^*(u) = u')}{f_{obv}(o_{u'} \,|\, \mathcal{K}_u, [t, t'], \sigma^*(u) = u')}. \tag{5.12}$$

Since the denominator does not depend on a_u, the optimisation problem reduces to maximising the nominator. This can be formulated as the global optimisation problem:

$$\arg\max_{s \in \mathcal{S}_u, \Xi^{s,o_{u'}}} \; \arg\max_{(t_b^1, t_e^1, \ldots, t_b^{N(s)}, t_e^{N(s)})} f_{actTraj,obv}(a_u, o_{u'} \,|\, \mathcal{K}_u, [t, t'], \sigma(u) = u') \tag{5.13}$$

where $a_u \equiv (\langle \psi_1^s, t_b^1, t_e^1 \rangle, \ldots, \langle \psi_{N(s)}^s, t_b^{N(s)}, t_e^{N(s)} \rangle)$ and $t_b^i \in \mathcal{T}_b^{i,s,o_{u'}}$ and $t_e^i \in \mathcal{T}_e^{i,s,o_{u'}}$ for all $1 \leq i \leq N(s)$. To solve this optimisation problem, first, for each $s \in \mathcal{S}_u$ we consider all of its possible decompositions. Then for each pair $(s, \Xi^{s,o_{u'}})$, we search for a sequence of entering and exiting time points that maximise the joint probability density function and record the largest probability density. To find the (approximate) optimum sequence, we can refer to a number of algorithms. We use *Simulated Annealing* in our implementation. Last, we choose the pair with the largest probability density and with the corresponding time sequence the optimum activity trajectory is thus constructed.

5.4.3 Discussion

The complexity of our tracking attack is mainly involved in the calculation of users' probability density of issuing an observed trajectory, (i.e., Equation 5.11) and solving the optimisation problem in the de-obfuscation attack (i.e., Equation 5.13). Both of them require to traverse all possible pairs of PoI sequences and decompositions. For each pair, in Equation 5.11, we calculate a multi-level integration while Equation 5.13 performs the algorithm of simulated annealing to calculate the most likely sequence of entering and exiting time points. In this thesis, we explore the Monte Carlo method to approximately calculate the values of the multi-level integrations. Since the time complexity of both simulated annealing and the Monte Carlo method are polynomial, the computational overhead is mainly determined by the number of the pairs of PoI sequences and decompositions. This number actually depends on three factors: the length of the observation period, the number of visited PoIs and observed events. The observation period subsequently decides the maximum number of PoIs that can be visited because a minimum transition time is needed to accomplish a transition between PoIs. Let P be the maximum number of PoIs that can be visited in the time period $[t, t']$, M be the size of Ψ_u, Q be the number of observed events. Then the worst-case complexity is $\mathcal{O}((M \cdot Q)^P)$, which grows exponentially in P. In our validation,

we observe that on average a user visits four PoIs per day. For the LBSs such as check-ins, according to Foursquare, an active user on average issues 2.3 posts per day. Therefore, the computational overhead is still manageable due to the small numbers of visited PoIs and observed events. In order to run our attack with more PoIs and longer time periods, we resort to sampling methods which approximately estimate the target result with only part of the search space. Based on users' profiles, we can generate possible pairs of PoI sequences and decompositions, for example, by random walking. With carefully chosen number of sampled pairs and accuracy tolerance, we can still ensure a good approximation of the calculation.

5.5 Localisation Attack

Our mobility profile can be used not only to track users' activities but also to implement the general attacks addressed in the literature. In this section we take *localisation attack* as an example to illustrate the generosity of our model.

In this attack, given an observed trajectory and a time point in the time interval of the trajectory, the adversary aims to find the PoI at which the user was present at the given time point. The result of this attack is a probability distribution over the set of PoIs of the originator of the observed trajectory.

We assume that the anonymising strategy σ is already known to the adversary and that $\sigma(u) = u'$. Given an observed trajectory $o_{u'}$ and the time point $t'' \in [t, t']$, the adversary aims to calculate the probability that user u was at PoI $\psi \in \Psi_u$ at time t'', i.e.,

$$\Pr((\psi, t'') \,|\, \sigma(u) = u', o_{u'}, \mathcal{K}_u). \tag{5.14}$$

This probability can be calculated as the ratio of the joint probability density that user u generated $o_{u'}$ and was located at ψ at t'' over the probability density that $o_{u'}$ is generated by user u. Let $f_{obv,poi}((\psi, t''), o_{u'} \mid \mathcal{K}_u, \sigma(u) = u')$ be the joint probability density. Then we have

$$\Pr((\psi, t'') \,|\, \sigma(u) = u', o_{u'}, \mathcal{K}_u) = \frac{f_{obv,poi}((\psi, t''), o_{u'} \,|\, \mathcal{K}_u, \sigma(u) = u')}{f_{obv}(o_{u'} \,|\, \mathcal{K}_u, \sigma(u) = u')}. \tag{5.15}$$

The probability density function $f_{obv,poi}$ can be obtained by marginalising the joint probability densities of the activity and observed trajectories, i.e., $f_{actTraj,obv}$ over the subset of activity trajectories where user u is at ψ at t''. To find this subset of activity trajectories, we consider $\mathcal{S}_u^{\psi} \subseteq \mathcal{S}_u$ consisting of all sequences in $\mathcal{S}_u$ that contain ψ, i.e., $\mathcal{S}_u^{\psi} = \{s \mid s \in \mathcal{S}_u \wedge \psi \in s\}$. Let $o_{u'}^{t''}$ be the observed trajectory $o_{u'}$ with added a dummy observed event $\langle u', t'', \cdot \rangle$ if no observed event with the issuing time t'' is in $o_{u'}$. Then, for each $s \in \mathcal{S}_u^{\psi}$ we consider all decompositions of $o_{u'}^{t''}$ with respect to s where the observed event with issuing time t'' corresponds to ψ, i.e., $\Xi^{s,o_{u'}^{t''}}$ such that $\langle u', t'', \cdot \rangle \in \Xi_i^{s,o_{u'}^{t''}}$ and $\psi_i^s = \psi$. Let $\Xi^{s,o_{u'}^{t''},(\psi,t'')}$ denote such a decomposition. Since the dummy event is not one of the observed events, we should exclude it when the probability density of issuing observed trajectories. We define $\Xi_i^{s,o_{u'}^{t''},(\psi,t'')} / \langle u', t'', \cdot \rangle$ as the block with the dummy request removed. Then,

the probability density can be calculated as the following:

$$
\begin{aligned}
& f_{obv,poi}((\psi, t''), O \equiv o_{u'} \,|\, \mathcal{K}_u, \sigma(u) = u') \\
= & \sum_{s \in \mathcal{S}_u^{\psi}} \sum_{\Xi^{s,o_{u'}^{t''},(\psi,t'')}} \int_{\mathcal{T}_b^{1,s,O} \times \mathcal{T}_e^{1,s,O} \times \ldots \times \mathcal{T}_b^{N(s),s,O} \times \mathcal{T}_e^{N(s),s,O}} \\
& f_{actTraj}((\langle \psi_1^s, t_b^1, t_e^1 \rangle, \ldots, \langle \psi_{N(s)}^s, t_b^{N(s)}, t_e^{N(s)} \rangle) \,|\, \mathcal{K}_u) \cdot \\
& \prod_{i=1}^{N(s)} f_{issue}(\Xi_i^{s,O,(\psi,t'')} / \langle u', t'', \cdot \rangle \,|\, [t_b^i, t_e^i], \psi_i^s, \sigma(u) = u') \; \mathrm{d}t_e^{N(s)} \; \mathrm{d}t_b^{N(s} \; \ldots \mathrm{d}t_e^1 \; \mathrm{d}t_b^1 \; ,
\end{aligned}
\tag{5.16}
$$

where the second summation is over all possible decompositions satisfying the condition discussed above.

5.6 Validation

In this section, we pursue two goals: (i) constructing user mobility profiles and identifying their main features in users' real-life movements; (ii) evaluating the effectiveness of the implemented attacks using our new mobility profiles.

5.6.1 Constructing mobility profiles

We present our method that the adversary may adopt to construct users' mobility profiles by exploring users' travel history. Moreover, we implement this method and discuss the main features of users' mobility profiles with a real-life trajectory dataset. We start with the specification of the dataset.

We explore a real-life GPS trajectory dataset to justify our work, which is collected in the *Geolife* project of Microsoft Research Asia [ZWZ+08]. The dataset consists of 17,621 trajectories from 182 users in a period of over five years (from April 2007 to August 2012). Most of these users' movements took place in Beijing, China. Each trajectory corresponds to a user's movement in one day. The trajectories cover a total length of about 1,250,000 km and a total duration of more than 48,000 hours. Moreover, the trajectories are collected in a high frequency. Over 90% of the locations are recorded less than every 5 seconds. In our experiments, we select ten representative users based on the number of their collected trajectories. On average, each user has over 200 daily trajectories.

A user's mobility profile mainly consists of a set of PoIs (Ψ_u), a high-order Markov chain describing the movement between PoIs (Ω_u), a probability density function of the amount of time that the user stays at a PoI (Γ_s^u), a probability density function of the amount of time that the user spends on transiting between PoIs (Γ_t^u). We show how to extract such information from a trajectory dataset.

Extracting Ψ_u. In the literature, two types of methods have been identified to obtain a user's PoIs: *static* or *dynamic* [GNPP07]. In static methods, PoIs are obtained by referring to public information. For instance, places such as schools and bus stops are labelled with different icons in Google Maps. However, static methods can only identify the PoIs that attract common interest of users while

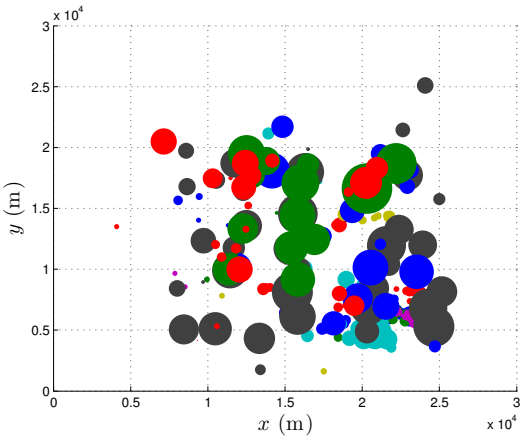


Figure 5.3: The distribution of users' PoIs.

those personal PoIs, e.g., meeting places with friends, are still uncovered. Dynamic methods offer a way to discover a user's PoIs from his travel history based on some heuristics such as short stay time and low speed [URT11]. Attackers should combine these two types of methods in to construct a complete PoI set.

In our validation, we adopt the method proposed by Chen et al. [CPX13] to dynamically compute PoIs. This method explores the heuristic that a PoI is usually a small region where a user tends to stay for certain amount of time. The idea is to first calculate *stay points* representing the regions that a user stayed during his movements and then make use of hierarchical clustering to cluster close stay points. The smallest region that covers all stay points in a cluster forms a PoI.

In Figure 5.3 we plot the calculated PoIs of the ten users labelled by different colours. We use filled circles to represent PoIs with the same area as that of the original ones. We can see that these ten users travelled in an area of $30km \times 30km$. The average area of these PoIs is 0.317 km^2 and each user has 28 PoIs on average. In the experiments of Shokri et al. [STBH11], they partition the San Francisco Bay area into a grid of 40 cells which is similar to our area. However, each cell has an area of 23 km^2, which is much larger than the PoIs we have constructed. Therefore, in our framework using PoIs we can describe users' positions in a higher precision, which also leads to fewer states in the Markov chains to capture users' movements among PoIs.

Constructing Ω_u. The three elements of Ω_u (see Definition 5.1) need to be extracted for activity trajectories. Thus we start with computing activity trajectories based on users' travel history. A user u's travel history H_u^{or} records his whereabouts in a past time period and has the following form $(\langle t_1, r_1\rangle, \ldots, \langle t_m, r_m\rangle)$, where r_i is the user's position at t_i $(1 \leq i \leq m)$. We remove the identity u from the sequence as it is clear from the context. Our main idea of calculating activity trajectories is to extract the subsequences of H_u^{or} that are contained in some PoIs as they capture user u's stay in PoIs. Given a subsequence $ps = (\langle t_{x_1}, r_{x_1}\rangle, \ldots, \langle t_{x_m}, r_{x_m}\rangle)$ of H_u^{or}, it is called a *PoI segment* if it is a *maximal* subsequence restrained in a PoI. Formally,

$$\exists \psi \in \Psi_u \left((\forall 1 \leq i \leq m, r_{x_i} \in \psi \wedge x_{i+1} = x_i + 1) \wedge (r_{x_1-1} \notin \psi \wedge r_{x_m+1} \notin \psi)\right).$$

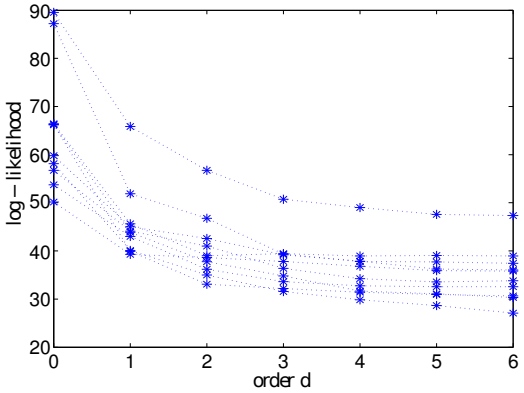


Figure 5.4: The log-likelihood of orders.

We use $poi(ps)$ to denote the PoI shared by all elements in ps and $t_{min}(ps)$ ($t_{max}(ps)$) to represent the minimum (maximum) time instances of ps, i.e., t_{x_1} (t_{x_m}). Let $(ps_1, \ldots, ps_k)$ be the sequence of all PoI segments in H_u^{or} in the ascending order of time. Then we can transform H_u^{or} to an activity trajectory as follows: $\forall 1 \leq i \leq k$,

$$\psi_i = poi(ps_i); \quad T_b^i = t_{min}(ps_i); \quad T_e^i = t_{max}(ps_i). \tag{5.17}$$

We then explore the maximum likelihood estimation method to calculate the transition matrix Ω_u. Given an order d, for each sequence of PoIs of length d, e.g., $\eta \in \Psi^d$, we maintain a counter $C_{\eta \to \psi}$ to record the number of occurrences of η before ψ. Thus, according to [CKRS12], we have

$$\Omega_u(\eta, \psi) = \frac{C_{\eta \to \psi}}{\sum_{\psi \in \Psi_u} C_{\eta \to \psi}}. \tag{5.18}$$

For a sequence of PoIs visited by a user, we can calculate the likelihood of the user to follow this sequence based on his transition matrix. If the transition matrix correctly captures users' transitions between PoIs, then the calculated probability will be larger. We use the logarithm of the probability, i.e., *log-likelihood*, to evaluate the quality of calculated matrices, formally, $-\sum_{i=d+1}^{x} \log \Pr(\psi_{y_i} \mid \psi_{y_1}, \ldots, \psi_{y_{i-1}})$. A smaller log-likelihood indicates a larger probability of generating the sequence. In Figure 5.4 we show the changes of the log-likelihood of the ten selected users, when different orders are used. Each point corresponds to an average of the log-likelihood of 10 sequences of length 20. It is obvious that larger orders increase the prediction accuracy. However, the magnitude of improvement decreases when d is increased. Since a larger d will increase the space to store the Markov chains and give rise to more computation overhead, the value of d should be decided carefully to balance accuracy and computational cost. We choose the order of users' transition matrices based on the amount of decrease, in terms of log-likelihood, between consecutive order values. When the decrease is smaller than a threshold δ, we select the previous value as the order of the user's transition matrix. In Table 5.2 we show the number of users for different orders when δ is set to two and three, respectively.

Table 5.2: The # of users with transition matrices of different orders.

	$\delta = 2$				$\delta = 3$			
order	1	2	3	4	1	2	3	4
#users	1	2	5	2	2	5	3	0

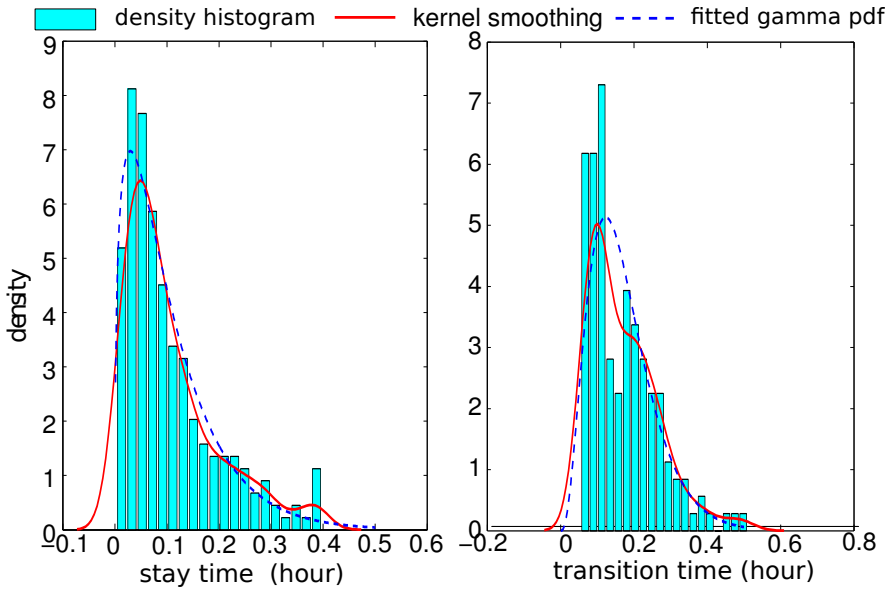


Figure 5.5: The probability density function of stay time and transition time.

Estimating Γ_s^u and Γ_t^u. We explore the methods of probability density estimation to calculate the fitted density functions for Γ_s^u and Γ_t^u. From the activity trajectory calculated above, we can extract all the occurrences of stay time of the user at a given PoI as well as the transition time between PoIs. In this chapter, we apply Gaussian kernel smoothing method.

For the selected users in the *Geolife* dataset, we cannot collect sufficient samples for *all PoIs* of the user and *all transitions* between PoIs because (i) users did not visit all their PoIs frequently enough; (ii) the data collection period is not long enough. To illustrate, we choose the user 153 in the dataset and plot the density histogram of his stay time in a PoI and transition time spent on transition between two PoIs in Figure 5.5. The corresponding estimated density functions are also presented. We can observe that the shape of the estimated functions are similar to Gamma distributions. We then fit Gamma distributions to the data and plot them with blue dashed curves. The fitted functions pass the Chi-square test, a classic test for the goodness-of-fit of a distribution to the sampled data, with p-values of 0.57 and 0.31, respectively. Notice that p-values range between 0 and 1 and indicates the fitness of distribution under hypothesis with the sample data.

5.6.2 Evaluating privacy attacks

We implement our attacks as explained in Section 5.4. In this section, we validate our new model for user mobility profiles and extended framework by showing the effectiveness of our attacks. As mentioned above, the *Geolife* dataset does not contain sufficient amount of data to allow us to extract complete mobility profiles

for all the selected users, especially for the parameters in the probabilistic density functions related to stay time and transition time. However, since we have learnt the types of such density functions as discussed in the previous section, we *partially* simulate users' profiles when the required information is not extractable. This will not impose much impacts on our validation since our target focuses on the effectiveness of the privacy attacks under the assumption of the availability of user mobility profiles. In the rest of this section, we start with describing the experimental setting before showing the results.

Experimental setting. We need to set up our experiments from three perspectives: user mobility profile, activity trajectories and observed trajectories. We begin with user mobility profiles. For the ten selected users from the *Geolife* dataset, we extract their transition matrices whose orders have been shown in Table 5.2. For their stay time and transition time, we assume that the corresponding probability density functions follow Gamma distributions when there are not sufficient number of samples to extract them. By referring to the parameters of the fitted gamma distributions of stay time and transition time, we generate values of the *shape* and *scale* parameters for each of these unknown Gamma distributions. Specifically, given two PoIs, we ensure that the simulated function for the transition time between them has a mean value which is consistent with the average transition time of the owner. For the stay time in a PoI, we applied the similar principle. With respect to users' rates of issuing requests in PoIs, we randomly choose their values between zero and three. This means that users on average issue at most three requests within one hour, which is reasonable in reality. The minimum transition time between two PoIs is determined by the distance between the centres of the PoIs and the maximum allowed speed which is preliminarily set to $20km/hour$.

With respect to activity trajectories, for each user we choose 40 daily trajectories from the *Geolife* dataset which contain less than six PoIs. The time information of these trajectories are not consistent with user mobility profiles as part of user profiles are simulated rather than extracted directly from the dataset. Thus, we proceed to extract the sequences of PoIs in them and make use of the simulated user profiles to generate the amount of time that users spend in and between the PoIs. The time of the simulated activity trajectories spans between 2 to 12 hours depending on the number of PoIs involved.

Last, we generate the observed trajectories of the selected users as the *Geolife* dataset does not contain such users' behaviour, i.e., exposing their locations for LBSs. Given an activity trajectory, we proceed to generate the corresponding exposed trajectory based on the owner's request issuing rate. In order to analyse the influence of the number of issued requests on location privacy, for each activity trajectory, we generate two exposed trajectories with lengths of one and three, respectively. For obfuscating LPPMs, we implement a simple cloaking mechanism which reduces the precision of the coordinates of the locations in the exposed trajectories. In our experiments, we set two precisions, namely, 0.001 and 0.01, and examine the sensitivity of location privacy to the reduced precision. These two precisions enlarge a position to a region with area of about 0.02 to 2.25 km^2, respectively.

Location privacy metric. Recall that in our framework, we make use of the estimation error of the adversary to measure location privacy provided by LPPMs.

Recall that the de-anonymising attack calculates the most likely anonymising strategy. Given a set of observed trajectories, each of which corresponds to a unique user, we store the output of the attack as a map assigning each observed trajectory the most probable user identity. Thus, we can define the adversary's estimation error as the percentage of the user identities that are not assigned to the right observed trajectories. Let σ be the anonymising strategy deployed. Then Formally, the estimation error for de-anonymisation can be defined as follows:

$$privacy_{da}(o_u^{[t,t']}, \sigma^*) = \frac{|\{u \in \mathcal{U} \,|\, \sigma^*(u) \neq \sigma(u)\}|}{|\mathcal{U}|}. \tag{5.19}$$

In our tracking attack, for any observed trajectory of a user, its output is the most probable activity trajectory. Let $a_u^{[t,t']^*}$ be the calculated activity trajectory. Since there is only one estimated trajectory, according to Equation 5.1, $\Pr(a_u^{[t,t']^*} \,|\, o)$ is set to 1.0 meaning that the adversary is certain that this activity trajectory is the most likely from his view. Then the location privacy is determined by the distance between $a_u^{[t,t']^*}$ and user u's real activity trajectory $a_u^{[t,t']}$, i.e., $\| a_u^{[t,t']^*}, a_u^{[t,t']} \|$. Intuitively, if we ignore the transitions between PoIS, or simply treat the positions during transition as a specific PoI, i.e., $\perp$, then two activity trajectories are equivalent if and only if two users are in the same PoIs at any time. This leads us to use the proportion of time when two users are not in the same PoI to define the distance between two activity trajectories. Let $PoI(a_u, t)$ returns the PoI where u stays at time t according to a_u. Then the metric can be defined as follows:

$$privacy_{do}(o_{u'}^{[t,t']}, a_u^{[t,t']^*}) = \frac{|\{t'' \in \mathcal{T} \,|\, PoI(a_u^{[t,t']^*}, t'') \neq PoI(a_u^{[t,t']}, t'')\}|}{t' - t}. \tag{5.20}$$

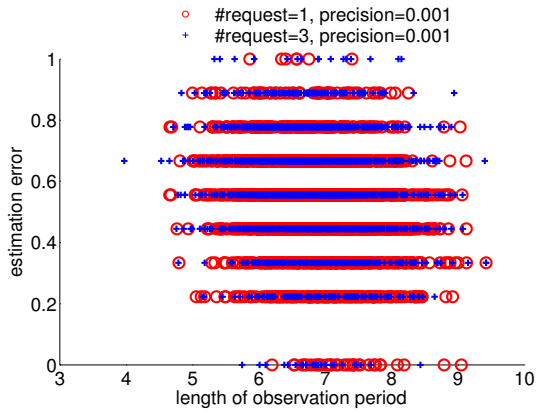
In our localising attack, the output is a probability distribution over the set of all PoIs. Then the estimation error is equivalent to the sum of the probabilities of the PoIs which are not the correct one. Suppose user u who is at PoI ψ at time t''. Formally, the adversary's estimation error of user u's position at t'' can be defined as the following:

$$privacy_{loc}(o_u^{[t,t']}, \langle \psi, t'' \rangle) = \sum_{\psi' \in \Psi_u / \psi} \Pr((\psi', t'') \,|\, \sigma(u) = u', o_{u'}^{[t,t']}, \mathcal{P}_u). \tag{5.21}$$

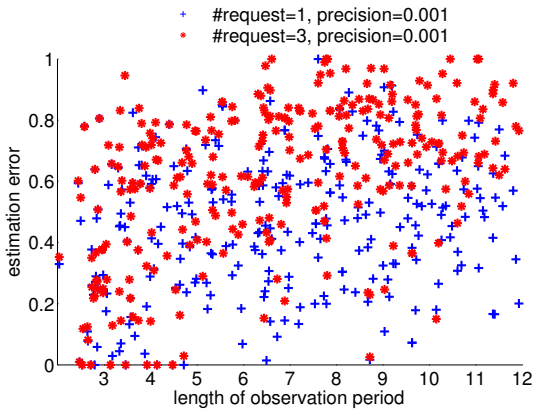
Experimental results. In this section, we present the experimental results of our attacks on location privacy from two aspects: their effectiveness and their sensitivity to distinctive factors. In the attacks, we assume that the adversary has access to the time periods when users travelled each of their activity trajectories, which are called *observation periods* in the following discussion.

We validate the de-anonymising attack by checking whether it can find the owners when a set of daily observed trajectories are learnt by the adversary. We call such a set of daily observed trajectory from the selected users a *daily observation*. To construct a daily observation, for each user we randomly choose one of his observed trajectories. We run our implementation on 5,000 daily observations, each

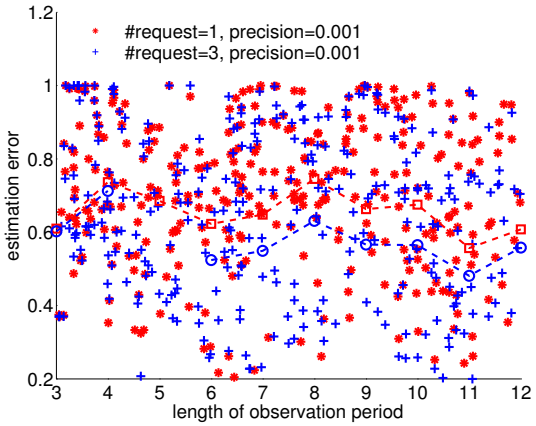
containing one observed trajectory for all of the users. The tracking and localising attacks aim to infer information of given observed trajectories whose owners have been already learnt. In our experiments, we apply our implementation of these two attacks on each observed trajectory of the selected users. Note that for the localising attack, for the sake of illustration and without loss of generality, we set the time point which the adversary is interested in about users' locations as the last time points of the observation periods.



(a) De-anonymisation attack.



(b) Tracking attack.



(c) Localising attack.

Figure 5.6: Estimation error vs. length of period.

Effectiveness. In Figure 5.6, we plot the adversary's estimation errors for all samples according to the length of observation periods when the reduced precision is set to 0.001, and in Table 5.3 we summarise the percentages of samples whose estimation errors fall into different intervals. Generally, most of the estimation errors range from 0.2 to 0.8. From Table 5.3, we can even see that for about 50% of the samples in our three attacks, the adversary has at least a probability of 0.4 to get the right target information. Therefore, from these statistics, we can conclude that our attacks are rather effective.

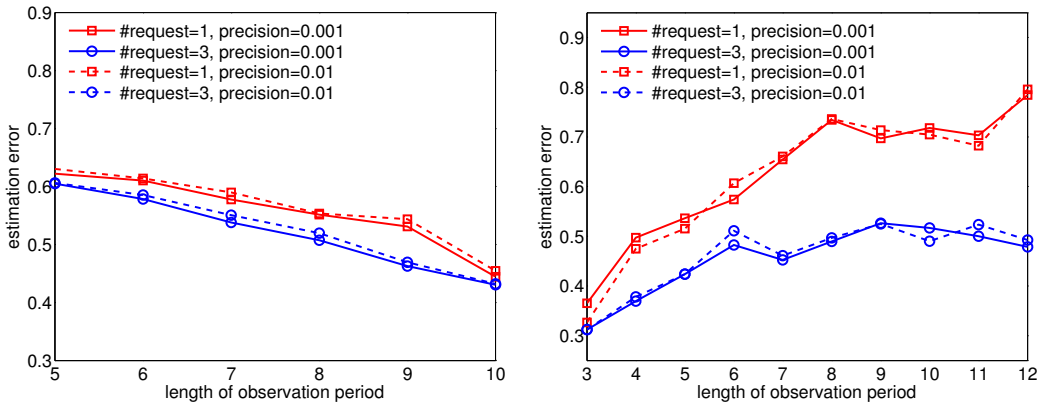
Sensitivity test. So as to have a comprehensive analysis, we study four major parameters that may have impact on users' location privacy: the number of issued LBS requests, the length of observation periods, the number of visited PoIs and the reduced precision.

Table 5.3: The distribution of estimation error.

Attacks	≤ 0.2	$0.2 - 0.4$	$0.4 - 0.6$	$0.6 - 0.8$	≥ 0.8
de-anonymising	0.01	0.16	0.46	0.32	0.04
tracking	0.11	0.18	0.26	0.31	0.14
localising	0.06	0.15	0.20	0.31	0.27

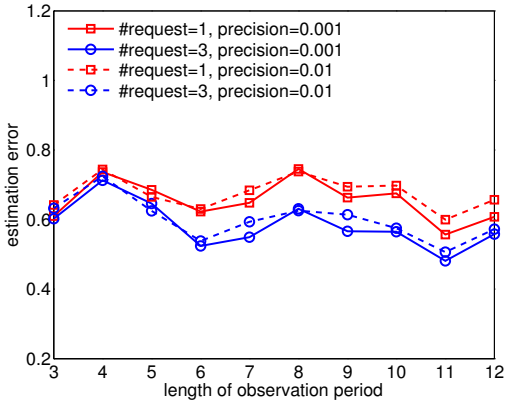
From Figure 5.6 we can observe that the adversary's estimation errors vary when different numbers of LBS requests are issued, i.e., more requests issued will lead to more privacy breached. The estimation errors when only one request is issued (annotated by red objects) are mainly located in the upper part of the sub-figures in Figure 5.6. This observation will be more visible when we discuss Figure 5.7 and Figure 5.8.

To test the sensitivity to the length of observation periods, in Figure 5.7 we group observed trajectories according to the length of their observation periods and depict the mean estimation error of the trajectories in each case. Specifically, we first put the observed trajectories with observation periods less than 3 hours in the first case and the rest are put into the ith cluster when their observation period is between $i-1$ hours and i hours.



(a) De-anonymising attack.

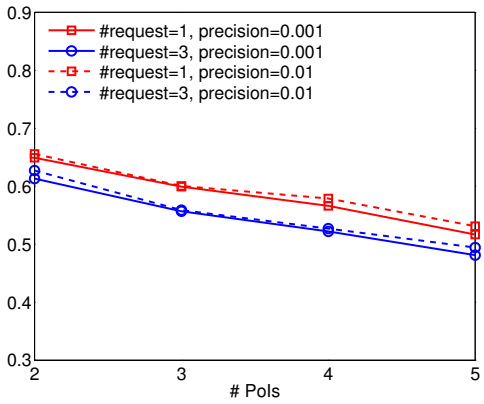
(b) Tracking attack.



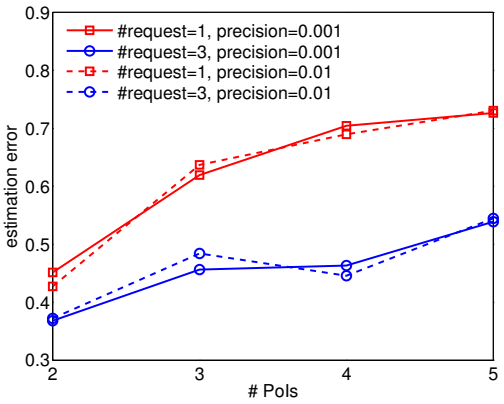
(c) Localising attack.

Figure 5.7: Mean estimation error vs. length of period.

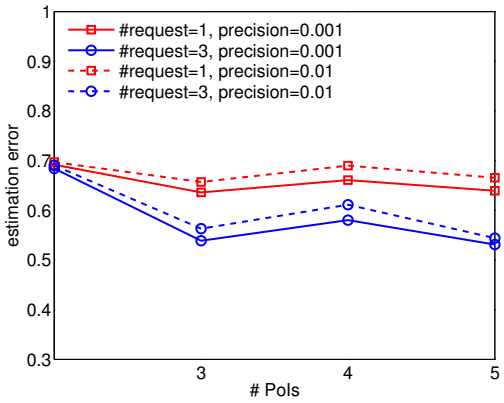
In general, our main observation is that the length of observation periods has



(a) De-anonymising attack.



(b) Tracking attack.



(c) Localising attack.

Figure 5.8: Mean estimation error vs. # PoIs.

different influence on the effectiveness of our attacks. In the de-anonymising attack, the mean estimation error decreases along with with the length of observation periods. This can be explained by the fact that users are more likely to travel distinctive trajectories in a longer time period. In other words, user mobility profiles can be better expressed in longer trajectories. In the tracking attack, we have the opposite observation. The estimation error increases significantly when users travel a longer time. This is because in a long period, users have more flexibility to arrange their visits to PoIs as well as the corresponding stay time in them. Compared to Figure 5.7(a) and 5.7(b), we can observe one difference in Figure 5.7(c) which shows the error changes in the localising attack. For users' daily activity trajectories the amount of travel time does not influence much the adversary's estimation on users' locations. This is because we make use of the higher order of Markov chains to model users' transition between PoIs, which leads to better prediction for the adversary. In addition, the consideration of users' patterns on stay time and transition time can help the adversary to infer the right number of visited PoIs.

In Figure 5.8 we present the changes of the mean estimation error when users visit different numbers of PoIs. We can see that in all our three attacks, the number of visited PoIs has a similar impact to that of the length of observation periods. This is because of the fact that a longer period indicates more PoIs that can be visited.

In Figure 5.7 and 5.8, we also show the results when different reduced precisions are used. We can see that the increase of reduced precision from 0.001 to 0.01 does not have a visible improvement for users' location privacy. This is because the exploration of PoIs already improves the adversary's uncertainty and eliminates the impact of reduced precision, especially when the precision is not reduced significantly enough to counter the effect of PoIs known to the adversary.

From the above analysis, we conclude that our implementation of location privacy attacks can ensure the adversary with high probabilities to learn the correct information about users' movement. This demonstrates that our framework is expressive and our attacks are effective. In addition, we also conclude that users should be cautious with their location privacy when enjoying the convenience of LBSs through a comprehensive analysis on the sensitivity of our attacks to different factors.

5.7 Related Work

It is widely recognised that the exposure of locations can threaten users' privacy. We briefly present the state-of-the-art in location privacy protection and attack mechanisms.

LPPMs. In general, the LPPMs proposed in the literature can be divided into two types: *cryptography-based* and *non-cryptography-based*. The former type of LPPMs encrypt time-stamped requests to hide the involved spatio-temporal information from attackers and LBS providers [KSSM11]. Non-cryptography-based LPPMs aim to protect location privacy by modifying LBS requests before sending them to LBS providers. We can further categorise them into two classes – *anonymisation* and *obfuscation*. Anonymising LPPMs breaks the link between users and their locations by replacing users' identities with pseudonyms. Obfuscating LPPMs modifies the spatial information in LBS requests to increase the adversary's uncertainty. Cloaking [LOYK11, WXH$^+$12] and perturbation [MC09] are two of the most used obfuscating methods. The former reduces the precision of locations while the later adds other locations as noise. Request hiding and dummy adding [MC09] are another two obfuscating methods in which some requests are eliminated or issued as dummies. In this chapter, we focus on non-cryptography-based LPPMs due to its popularity in location-based applications such as geo-social networks.

Location privacy attacks & user profiles. In spite of the protection of LPPMs, location privacy is still at stake especially when user profiles are extracted and explored by the adversary. Mobility profiles are the most widely discussed user profiles. So far, many models for user mobility profiles have been proposed and we can categorise them into three types: *inertia-based* [SJLL00], *Markov chain-based* [STT$^+$12, AJN$^+$13] and *trajectory pattern-based* [MPTG09, CPX13]. Inertia-based models extract users' speed and direction to calculate future locations. One drawback of this type of models is that they do not work well in the prediction of long-term movements. Markov chain-based models capture the dependence of users' destinations on previous moves while trajectory pattern-based models extract the frequently visited sequences of PoIs. These models have one defect in

common that they do not *explicitly* consider the temporal information in user mobility profiles, such as stay time at a PoI, as part of the models. Users also form certain patterns with regard to requesting LBSs. Take request issuing time as an example. In continuous LBSs users periodically issue requests and the exposure time is thus available *a priori*. They are usually modelled as a set of discrete time points [STBH11]. In order to model sporadic LBSs, Shokri et al. [STD+11] assign a probability to each possible exposure time point representing the likelihood that a user requests LBSs at that particular time.

Formal frameworks. Due to the diversity of user profiles and location privacy attacks, a unified framework is needed to evaluate LPPMs. Shokri et al. [STBH11] make the first significant attempt to construct such a framework. Their framework provides means to formalise location privacy attacks and quantify location privacy by the expected estimation error of the adversary. Later this framework is extended and explored in many ways. It is adopted in [STT+12] to optimise the values for the parameters of LPPMs against strategic attackers who know the LPPM implementation and user mobility profiles. Herrmann et al. [AJN+13] refine this work by considering the bandwidth constraints when dummy requests are issued to perturb the real requests. In this chapter, we adopt and extend this framework to formalise our new tracking attack.

5.8 Conclusion

We proposed and implemented a new tracking attack with the aim to provide the adversary means to breach users' activity privacy. Other attacks on location privacy in [STBH11] can be formalised in our framework as well [CMP14]. Namely, the places where users visits and the entering and stay time of these places can be obtained through our attack in a direct way. To implement this attack, we proposed a new model for user profiles. Compared to existing works on user mobility profiles, our model can describe users' patterns with respect to mobility and requesting LBS in continuous time which is rather expressive. By making use of PoIs, our attack has a reasonable efficiency and can be extended to cover more general cases with little loss of accuracy.

Part IV

Concluding Remarks

6

Conclusion and Future Work

6.1 Conclusion

This thesis studied two security requirements in location-based services: location assurance and privacy. Location assurance is related to the trustworthiness of users' locations. Privacy is about the protection of the information contained in LBS requests, namely, locations and queries.

Location assurance is threatened by spoofing attacks on GNSS systems. In these attacks, receivers are fooled to calculate different locations from where they are actually located. Although eliminating spoofing is impractical and infeasible in the near future, our research shows that users can detect spoofing by evaluating the integrity of received GNSS signals. This evaluation subsequently helps a user to determine the extent to which he can trust his locations. Users' privacy is threatened in LBSs because queries and locations reveal their personal information. Although many privacy preserving methods are proposed, our research showed that users' privacy is still at risk. For query privacy, we presented a unified framework which can analyse the impact of different types of contextual information on query privacy. For location privacy, we present an attack that derives users' movement information from locations protected by existing location privacy preserving methods (LPPM). Such movement information can be used to effectively infer users' activities.

The research questions that were raised in the introduction chapter have been answered in Chapter 3, Chapter 4 and Chapter 5, respectively.

Research question 1: *How can we access the integrity of GNSS signals?*

To answer question 1, we made use of existing spoofing detection methods based on signal integrity evaluation. We identified three problems with them and proposed a trust framework to solve these problems. First, the reasoning is incorrectly implemented in spoofing detection methods to reach a conclusion on signal integrity from evidences. We clarified that it is *abductive* rather than *abductive*. Second, uncertainty is inevitable by nature due to the unpredicted influence from environment and the unknown ability of the adversary in tuning GNSS signals. However it is ignored in existing methods. We explored subjective logic to explicitly capture the uncertainty and took it into account when evaluating signal integrity. Third, due to the different evidences used in spoofing detection methods, conflicting conclusions exist. We designed three algorithms to resolve the conflict and generate an overall evaluation of the integrity of received GNSS signals.

Research question 2: *How can we protect users' query privacy confronting the*

adversary with various contextual information?

To answer this question, we proposed a general approach which can be used to analyse the threat to query privacy caused by the increasing amount of available contextual information. We proposed a formal framework to achieve this goal. In this framework, we divided contextual information into static and dynamic and implemented our framework with one type of contextual information for each of them, i.e., user profiles and query dependency. With a probabilistic distribution as the result of the adversary's attack, we defined a series of metrics based on information theory. These metrics can not only be used to measure users' query privacy provided by protecting methods but also allow users to express their requirements on query privacy flexibly and precisely. Based on our framework and metrics, we developed new area generalisation algorithms which protect query privacy according to users' requirements.

Research question 3: *How can we formally capture the threat to users' location privacy which targets at users' activities?*

To answer this question, we formally define an attack which targets at a new form of users' movement: activity trajectories. An activity trajectory provides sufficient information to infer a user's activities as well as the temporal information of each activity, i.e., beginning time and ending time. To perform this attack, we proposed a new model for user profiles, which models two types of user behaviour: moving and LBS requesting. Based on a real-life trajectory dataset, we validated our model for user mobility profiles. Through experiments, we showed that our attack and new model for user profiles ensure a large chance for the adversary to learn the correct activity trajectories.

6.2 Future Work

There are a few related research questions that deserve exploration but have not been addressed. We present two of them in this section.

6.2.1 Adding behaviour perturbation

In Chapter 5, we proposed a model for user profiles which captures users' patterns with respect to movement and requesting LBSs. The model effectively helps attackers infer users' activities based on their exposed locations in LBSs. However, there is an implicit assumption that users do not deviate from their 'normal' behaviour. In other words, they act as their profiles suggest. We did not take into account the cases when users deviate from their normal behaviour, i.e., *behaviour perturbation*. For instance, a user can go to a new restaurant which he has never been to because of the suggestion of a friend. It is also quite normal that users' agendas are interrupted by sudden incidents with higher priorities. Therefore, a future direction of research is to add users' behaviour perturbation into our model so that to capture users' behaviour in a more realistic way.

6.2.2 Adding dependency between users

In Chapter 4 and Chapter 5, we have a common assumption that users are independent of each other when they request LBSs and travel. This assumption is too strong for practical scenarios. Consider that Alice is a close friend of Bob and they usually hang out together. If we learn that Bob is at a party on Thursday night, then we can infer that Alice is also at the part with a high confidence.

There have already been a few works addressing this problem [CBC+10, SH12, OHSH14]. Shokri et al. take into account users' co-location information posted on social networks (e.g., photos) and calculate users' locations based on both users' mobility profiles and the locations where users are co-located. These studies suggest an interesting research direction which makes uses of the influence of friendship on a user's movement. Social networks such as Facebook, LinkedIn and Foursquare provide us sources to learn a user's friends and the patterns with respect to their interactions. A promising way to advance our research is to classify a user's friends according to the types of activities which they usually perform with the user, e.g., entertainment and sports. Although mobile data have been studied as subsidiary information to infer users' friendship networks [EPL09], more research is needed to study the impact of users' friends on their mobility.

Bibliography

[AAB+00] Dan Ariely, Wing Tung Au, Randall H. Bender, David V. Budescu, Christiane B. Dietz, Hongbin Gu, Thomas S. Wallsten, and Gal Zauberman. The effects of averaging subjective probability estimates between and within judges. *Journal of Experimental Psychology: Applied*, 6:130–147, 2000.

[Age13] European Global Navigation Satellite Systems Agency. GSA market report 2013. Avaible at `http://www.gsa.europa.eu/market/market-report`, October 2013.

[AJN+13] Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel. Optimal sporadic location privacy preserving systems in presence of bandwidth constraints. In *Proc. 12th ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 167–178. ACM Press, 2013.

[Ber05] Alastair R. Beresford. *Location Privacy in Ubiquitous Computing*. PhD thesis, University of Cambridge, January 2005.

[BHV07] Levente Buttyán, Tamás Holczer, and István Vajda. On the effectiveness of changing pseudonyms to provide location privacy in vanets. In *Proc. 4th European Workshop on security and privacy in ad-hoc and sensor networks (ESAS)*, volume 4572 of *Lecture Notes in Computer Science*, pages 129–141. Springer, 2007.

[BKH08] Paolo Bellavista, Axel Küpper, and Sumi Helal. Location-based services: Back to the future. *IEEE Pervasive Computing*, 7(2):85–89, 2008.

[BLPW08] Bhuvan Bamba, Ling Liu, Péter Pesti, and Ting Wang. Supporting anonymous location queries in mobile environments with PrivacyGrid. In *Proc. 17th International Conference on World Wide Web (WWW)*, pages 237–246. ACM Press, 2008.

[BMW+09] Claudio Bettini, Sergio Mascetti, Xiaoyang Sean Wang, Dario Freni, and Sushil Jajodia. Anonymity and historical-anonymity in location-based services. In *Proc. 1st International Workshop on Privacy in Location-based applications (PiLBA)*, volume 5599 of *Lecture Notes in Computer Science*, pages 1–30. Springer, 2009.

[Bor07] Kar Borre. *A Software-Defined GPS and Galileo Receiver*. Applied and Numerical Harmonic Analysis, 2007.

[Bri02] Thomas Brinkhoff. A framework for generating network-based moving objects. *GeoInformatica*, 6(2):153–180, 2002.

[BW93] Fergus Bolger and George Wright. Coherence and calibration in expert probability judgement. *Omega*, 21(6):629–644, 1993.

[Car03] James V. Carroll. Vulnerability assessment of the U.S. transportation infrastructure that relies on the global positioning system. *The Journal of Navigation*, 56(2):185–193, 2003.

[CBC+10] David J. Crandall, Lars Backstrom, Dan Cosley, Siddharth Suri, Daniel Huttenlocher, and Jon Kleinberg. Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences (PNAS)*, 107(52):22436–22441, 2010.

[CCL12] Xihui Chen, Harpes Carlo, and Gabriele Lenzini. Location assurance framework. Luxembourgish patent No. LU92003, August 2012.

[CFP12] Xihui Chen, David Fonkwe, and Jun Pang. Post-hoc user traceability analysis in electronic toll pricing systems. In *Proc. 7th and 5th International Workshop on Data Privacy Management and Autonomous Spontaneous Security (DPM/SETOP)*, volume 7731 of *Lecture Notes in Computer Science*, pages 29–42. Springer, 2012.

[CHL+13a] Xihui Chen, Carlo Harpes, Gabriele Lenzini, Miguel Martins, Sjouke Mauw, and Jun Pang. Demonstrating a trust framework for evaluating GNSS signal integrity. In *Proc. 20th ACM Conference on Computer and Communications Security (CCS)*, pages 1329–1332. ACM Press, 2013.

[CHL+13b] Xihui Chen, Carlo Harpes, Gabriele Lenzini, Sjouke Mauw, and Jun Pang. Location assurance and privacy in GNSS navigation. *ERCIM News*, 2013(94), 2013.

[CKLP14] Xihui Chen, Piotr Kordy, Ruipeng Lu, and Jun Pang. MinUS: Mining user similarity with trajectory patterns. In *Proc. 7th European Conference on Machining Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science. Springer, 2014. To appear.

[CKRS12] Flavio Chierichetti, Ravi Kumar, Prabhakar Raghavan, and Tamás Sarlós. Are web users really markovian? In *Proc. 21st World Wide Web Conference (WWW)*, pages 609–618. ACM Press, 2012.

[CLM+13] Xihui Chen, Gabriele Lenzini, Miguel Martins, Sjouke Mauw, and Jun Pang. A trust framework for evaluating GNSS signal integrity. In *Proc. 26th IEEE Security Foundations Symposium (CSF)*, pages 179–192. IEEE Computer Society, 2013.

[CLMP12] Xihui Chen, Gabriele Lenzini, Sjouke Mauw, and Jun Pang. A group signature based electronic toll pricing system. In *Proc. 7th International Conference on Availability, Reliability and Security (ARES)*, pages 85–93. IEEE Computer Society, 2012.

[CLMP13] Xihui Chen, Gabriele Lenzini, Sjouke Mauw, and Jun Pang. Design and formal analysis of a group signature based electronic toll pricing system. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, 4(1):55–75, 2013.

[CLMP14] Xihui Chen, Ruipeng Lu, Xiaoxing Ma, and Jun Pang. Measuring user similarity with trajectory patterns: Principles and new metrics. In *Proc. 16th Asia-Pacific Web Conference (APWeb)*, Lecture Notes in Computer Science. Springer, 2014. To appear.

[CMA09] Chi-Yin Chow, Mohamed F. Mokbel, and Walid G. Aref. Casper*: Query processing for location services without compromising privacy. *ACM Transactions on Database Systems (TODS)*, 34(4):1–48, 2009.

[CMP14] Xihui Chen, Andrzej Mizera, and Jun Pang. Quantifying location privacy revisited: Preliminary report, 2014.

[CP12] Xihui Chen and Jun Pang. Measuring query privacy in location-based services. In *Proc. 2nd ACM Conference on Data and Application Security and Privacy (CODASPY)*, pages 49–60. ACM Press, 2012.

[CP13] Xihui Chen and Jun Pang. Exploring dependency for query privacy protection in location-based services. In *Proc. 3rd ACM Conference on Data and Application Security and Privacy (CODASPY)*, pages 37–48. ACM Press, 2013.

[CP14] Xihui Chen and Jun Pang. Protecting query privacy in location-based services. *GeoInformatica*, 18(1):95–133, 2014.

[CPHL07] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. Efficient and robust pseudonymous authentication in vanet. In *Proc. 4th International Workshop on Vehicular Ad-hoc Networks (VANET)*, pages 19–28. ACM Press, 2007.

[CPX13] Xihui Chen, Jun Pang, and Ran Xue. Constructing and comparing user mobility profiles for location-based services. In *Proc. 28th Annual ACM Symposium on Applied Computing (SAC)*, pages 261–266. ACM Press, 2013.

[CPX14] Xihui Chen, Jun Pang, and Ran Xue. Constructing and comparing user mobility profiles. *ACM Transactions on the Web (TWEB)*, 2014. To appear.

[CZBP06] Reynold Cheng, Yu Zhang, Elisa Bertino, and Sunil Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *Proc. 6th International Workshop on Privacy Enhancing Technologies (PET)*, volume 4258 of *Lecture Notes in Computer Science*, pages 393–412. Springer, 2006.

[DL01] Luc Devroye and Gabor Lugosi. *Combinatorial Methods in Density Estimation*. Springe, 2001.

[DRRW10a] Rinku Dewri, Indrakshi Ray, Indrajit Ray, and Darrell Whitley. On the formation of historically k-anonymous anonymity sets in a continuous LBS. In *Proc. 6th International Conference on Security and Privacy in Communication Networks (SecureComm)*, volume 50 of *Lecture Notes in Computer Science*, pages 71–88. Springer, 2010.

[DRRW10b] Rinku Dewri, Indrakshi Ray, Indrajit Ray, and Darrell Whitley. Query m-invariance: Preventing query disclosures in continuous location-based services. In *Proc. 11th International Conference on Mobile Data Management (MDM)*, pages 95–104. IEEE Computer Society, 2010.

[DSCP03] Claudia Díaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In *Proc. 2nd International Workshop on Privacy Enhancing Technologies (PET)*, volume 2482 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 2003.

[EPL09] Nathan Eagle, Alex Pentland, and David Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences (PNAS)*, 106(36):15274–15278, 2009.

[FMHP09] Julien Freudiger, Mohammad Hossein Manshaei, Jean-Pierre Hubaux, and David C. Parkes. On non-cooperative location privacy: a game-theoretic analysis. In *Proc. 16th ACM Conference on Computer and Communications Security (CCS)*, pages 324–337. ACM Press, 2009.

[FSH09] Julien Freudiger, Reza Shokri, and Jean-Pierre Hubaux. On the optimal placement of mix zones. In *Proc. 9th International Symposium on Privacy Enhancing Technologies (PETS)*, volume 5672 of *Lecture Notes in Computer Science*, pages 216–234. Springer, 2009.

[GG03] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. 1st International Conference on Mobile Systems, Applications, and Services (MobiSys)*. USENIX Association, 2003.

[GHB08] Marta C. González, César A. Hidalgo, and Albert-László. Barabási. Understanding individual human mobility patterns. *Nature*, 453:779–782, 2008.

[GKK+08] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. Private queries in location based services: anonymizers are not necessary. In *Proc. the ACM SIGMOD International Conference on Management of Data*, pages 121–132. ACM Press, 2008.

[GKS07] Gabriel Ghinita, Panos Kalnis, and Spiros Skiadopoulos. PRIVE: anonymous location-based queries in distributed mobile systems. In *Proc. 16th International Conference on World Wide Web (WWW)*, pages 371–380. ACM Press, 2007.

[GL08] Bugra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing (TMC)*, 7(1):1–18, 2008.

[GNPP06] Fosca Giannotti, Mcro Nanni, Dino Pedreschi, and Fabio Pinelli. Mining sequences with temporal annotations. In *Proc. 21st ACM Symposium on Applied Computing (SAC)*, pages 593–597. ACM Press, 2006.

[GNPP07] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 330–339. ACM Press, 2007.

[Goo09] Michael F. Goodchild. *Location-based services*, 2009.

[HGXA07] Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *Proc. 14th ACM Conference on Computer and Communications Security (CCS)*, pages 161–171. ACM Press, 2007.

[HK00] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

[HLP+08] Todd E. Humphreys, Brent M. Ledvina, Mark L. Psiaki, Brady W. O'Hanlon, and Paul M. Kintner. Assessing the spoofing threat: Development of a portable GPS civilian spoofer. In *Proc. 21st Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS)*, pages 2314–2325. Institute of Navigation, 2008.

[HLY10] Lian Huang, Qingquan Li, and Yang Yue. Activity identification from gps trajectories using spatial temporal POIs' attractiveness. In *Proc. 2nd International Workshop on Location Based Social Networks (LBSN)*, pages 27–30. ACM Press, 2010.

[HMC+12] Carlo Harpes, Miguel Martins, Xihui Chen, Gabriele Lenzini, Sjouke Mauw, and Jun Pang. Implementation and validation of a localisation assurance service provider. In *Proc. 6th ESA Workshop on Satellite Navigation Technologies (NAVITEC)*, pages 1–8. IEEE Computer Society, 2012.

[Hum13] Todd Humphreys. Detection strategy for cryptographic GNSS antispoofing. *IEEE Transactions on Aerospace and Electronic Systems*, 49(2):1073–1090, 2013.

[Jay57a] Edwin Thompson Jaynes. Information theory and statistical mechanics. *Physical Review Series II*, 106(4):620–630, 1957.

[Jay57b] Edwin Thompson Jaynes. Information theory and statistical mechanics ii. *Physical Review Series II*, 108(2):171–190, 1957.

[JDR10] Audun Jøsang, Javier Diaz, and Maria Rifqi. Cumulative and averaging fusion of beliefs. *Information Fusion*, 11(2):192–200, 2010.

[JJBNL12] Ali Jafarnia-Jahromi, Ali Broumandan, John Nielsen, and Gérard Lachapelle. GPS vulnerability to spoofing threats and a review of antispoofing techniques. *International Journal of Navigation and Observation (IJNO)*, 2012, 2012.

[Jøs09] Audun Jøsang. Conditional reasoning with subjective logic. *Journal of Multiple-Valued Logic and Soft Computing (MVLSC)*, 15(1):5–38, 2009.

[Jøs12] Audun Jøsang. Subjective logic (book draft). Available at `http://folk.uio.no/josang/papers/subjective_logic.pdf`, 2012.

[JPD05] Audun Jøsang, Simon Pope, and Milan Daniel. Conditional deduction under uncertainty. In *Proc. 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*, volume 3571 of *Lecture Notes in Computer Science*, pages 824–835. Springer, 2005.

[KGMP07] Panos Kalnis, Gabriel Ghinita, Kyriakos Mouratidis, and Dimitris Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(12):1719–1733, 2007.

[KH05] Elliott D. Kaplan and Christopher Hegarty. *Understanding GPS: Principles and Applications (second edition)*. Artech House, 2005.

[Kru07] John Krumm. Inference attacks on location tracks. In *Proc. 5th International Conference on Pervasive Computing (PERVASIVE)*, volume 4480 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2007.

[KSSM11] Ali Khoshgozaran, Cyrus Shahabi, and Houtan Shirani-Mehr. Location privacy: going beyond k-anonymity, cloaking and anonymizers. *Knowledge and Information Systems (KAIS)*, 26(3):435–465, 2011.

[Kuh04] Markus G. Kuhn. An asymmetric security mechanism for navigation signals. In *Proc. 6th Workshop on Information Hiding (IH)*, volume 3200 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 2004.

[KYS05] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. Protection of location privacy using dummies for location-based services. In *Proc. 21st International conference on Data Engineering (ICDE)*, pages 12–48. IEEE Computer Society, 2005.

[LFK05] Lin Liao, Dieter Fox, and Henry A. Kautz. Location-based activity recognition using relational markov networks. In *Proc. 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 773–778. IJCAI/AAAI, 2005.

[LLV07] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proc. 23rd International Conference on Data Engineering (ICDE)*, pages 106–115. IEEE Computer Society, 2007.

[LOYK11] Byoungyoung Lee, Jinoh Oh, Hwanjo Yu, and Jong Kim. Protecting location privacy using location semantics. In *Proc. 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1289–1297. ACM Press, 2011.

[LV03] Peter Lyman and Hal R. Varian. How much information. Available at `http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/`, 2003.

[Mac67] James B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[MBB13] Sergio Mascetti, Letizia Bertolaja, and Claudio Bettini. A practical location privacy attack in proximity services. In *Proc. 14th IEEE International Conference on Mobile Data Management (MDM)*, pages 87–96. IEEE Computer Society, 2013.

[MBFW07] Sergio Mascetti, Claudio Bettini, Dario Freni, and X. Sean Wang. Spatial generalization algorithms for LBS privacy preservation. *Journal of Location Based Services (JLBS)*, 1(3):179–207, 2007.

[MC09] Joseph T. Meyerowitz and Romit Roy Choudhury. Realtime location privacy via mobility prediction: creating confusion at crossroads. In *Proc. 10th Workshop on Mobile Computing Systems and Applications*. ACM Press, 2009.

[MCA07] Mohamed F. Mokbel, Chi-Yin Chow, and Walid G. Aref. The new casper: A privacy-aware location-based database server. In *Proc. 23rd International Conference on Data Engineering (ICDE)*, pages 1499–1500. IEEE Computer Society, 2007.

[MHL09] Paul Y. Montgomery, Todd E. Humphreys, and Brent M. Ledvina. Receiver-autonomous spoofing detection: Experimental results of a multi-antenna receiver defense against a portable civil GPS spoofer. In *Proc. 22th Technical Meeting of The Institute of Navigation*, pages 124–130, 2009.

[MKGV07] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), 2007.

[MPTG09] Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 637–646. ACM Press, 2009.

[MS99] Chris Manning and Hinrich Schuütze. *Foundations of Statistical Natural Language Processing*. Cambridge, 1999.

[NBL10] John Nielsen, Ali Broumandan, and Gérard Lachapelle. Spoofing detection and mitigation. *GPS World*, pages 27–33, September 2010.

[NLD+12] Tyler Nighswander, Brent M. Ledvina, Jonathan Diamond, Robert Brumley, and David Brumley. GPS software attacks. In *Proc. 19th ACM Conference on Computer and Communications Security (CCS)*, pages 450–461. ACM Press, 2012.

[OHSH14] Alexandra Mihaela Olteanu, Kevin Huguenin, Reza Shokri, and Jean-Pierre Hubaux. Quantifying the effect of co-location information on location privacy. In *Proc. 14th Privacy Enhancing Technologies Symposium (PETS)*, 2014. To appear.

[PCDC10] Oscar Pozzobon, Luca Canzian, Matteo Danieletto, and Andrea Dalla Chiara. Anti-spoofing and open GNSS signal authentication with signal authentication sequences. In *Proc. 5th ESA Workshop on Satellite Navigation Technologies (NAVITEC)*. IEEE Computer Society, 2010.

[PJ08] Panagiotis Papadimitratos and Aleksandar Jovanovic. GNSS-based positioning: Attacks and countermeasures. In *Proc. IEEE Military Communications Conference (MILCOM)*, pages 1–7. IEEE Computer Society, 2008.

[PL11] Balaji Palanisamy and Ling Liu. MobiMix: Protecting location privacy with mix-zones over road networks. In *Proc. 27th International Conference on Data Engineering (ICDE)*, pages 494–505. IEEE Computer Society, 2011.

[POB+13] Mark Psiaki, Brady O'Hanlon, Jahshan Bhatti, Daniel Shepard, and Todd Humphreys. GPS spoofing detection via dual-receiver correlation of military signals. *IEEE Transactions on Aerospace and Electronic Systems (TAES)*, 49(4):2250 – 2267, 2013.

[Pri13] Bogdan Pricope. *Positioning using terrestrial wireless systems*. PhD thesis, Jacobs University, 2013.

[PSR+13] Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady L. Andrienko, Natalia V. Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, José Antônio Fernandes de Macêdo, Nikos Pelekis, Yannis Theodoridis, and Zhixian

Yan. Semantic trajectories modeling and analysis. *ACM Computing Surveys (CSUR)*, 45(4):42, 2013.

[RMPADF13] David Rebollo-Monedero, Javier Parra-Arnau, Claudia Díaz, and Jordi Forné. On the measurement of privacy as an attacker's estimation error. *International Journal of Information Security (IJIS)*, 12(2):129–149, 2013.

[RPB09] Daniele Riboni, Linda Pareschi, and Claudio Bettini. Privacy in georeferenced context-aware services: A survey. In *Proc. 1st International Workshop on Privacy in Location-Based Applications (PiLBA)*, volume 5599 of *Lecture Notes in Computer Science*. Springer, 2009.

[RPBJ09] Daniele Riboni, Linda Pareschi, Claudio Bettini, and Sushil Jajodia. Preserving anonymity of recurrent location-based queries. In *Proc. 16th International Symposium on Temporal Representation and Reasoning (TIME)*, pages 62–69. IEEE Computer Society, 2009.

[RR98] Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security (TISS)*, 1(1):66–92, 1998.

[Sam01] Pierangela Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 13(6):1010–1027, 2001.

[SAV08] Heechang Shin, Vijayalakshmi Atluri, and Jaideep Vaidya. A profile anonymization model for privacy in a personalized location based service environment. In *Proc. 9th International Conference on Mobile Data Management (MDM)*, pages 73–80. IEEE Computer Society, 2008.

[SAV11] Heechang Shin, Vijayalakshmi Atluri, and Jaideep Vaidya. A profile anonymization model for location-based services. *Journal of Computer Security (JCS)*, 19(5):795–833, 2011.

[SD03] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *Proc. 2nd International Workshop on Privacy Enhancing Technologies (PET)*, volume 2482 of *Lecture Notes in Computer Science*, pages 41–53. Springer, 2003.

[SH12] Mudhakar Srivatsa and Mike Hicks. Deanonymizing mobility traces: using social network as a side-channel. In *Proc. 19th ACM Conference on Computer and Communications Security (CCS)*, pages 628–637. ACM Press, 2012.

[SHSH11] Francisco Santos, Mathias Humbert, Reza Shokri, and Jean-Pierre Hubaux. Collaborative location privacy with rational users. In *Proc. 2nd International Conference on Decision and Game Theory*

for Security (GameSec), volume 7037 of *Lecture Notes in Computer Science*, pages 163–181. Springer, 2011.

[SJLL00] Simonas Saltenis, Christian S. Jensen, Scott T. Leutenegger, and Mario A. Lopez. Indexing the positions of continuously moving objects. In *Proc. 16th ACM SIGMOD Conference on Management of Data (SIGMOD)*, pages 331–342. ACM Press, 2000.

[STBH11] Reza Shokri, Georgios Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying location privacy. In *Proc. 32nd IEEE Symposium on Security and Privacy (S&P)*, pages 247–262. IEEE Computer Society, 2011.

[STD+10] Reza Shokri, Carmela Troncoso, Claudia Díaz, Julien Freudiger, and Jean-Pierre Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *Proc. 2010 ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 115–118. ACM Press, 2010.

[STD+11] Reza Shokri, George Theodorakopoulos, George Danezis, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. Quantifying location privacy: The case of sporadic location exposure. In *Proc. 11th International Symposium on Privacy Enhancing Technologies (PETS)*, volume 6794 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2011.

[STT+12] Reza Shokri, George Theodorakopoulos, Carmela Troncoso, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. Protecting location privacy: optimal strategy against localization attacks. In *Proc. 19th ACM Conference on Computer and Communications Security (CCS)*, pages 617–627. ACM Press, 2012.

[TLM09] Kar Way Tan, Yimin Lin, and Kyriakos Mouratidis. Spatial cloaking revisited: Distinguishing information leakage from anonymity. In *Proc. 11th International Symposium on Spatial and Temporal Databases (SSTD)*, volume 5644 of *Lecture Notes in Computer Science*, pages 117–134. Springer, 2009.

[TPRC11] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. On the requirements for successful GPS spoofing attacks. In *Proc. 18th ACM Conference on Computer and Communications Security (CCS)*, pages 75–86. ACM Press, 2011.

[URT11] Muhammad Reaz Uddin, Chinya V. Ravishankar, and Vassilis J. Tsotras. Finding regions of interest from trajectory data. In *Proc. 12th IEEE International Conference on Mobile Data Management (MDM)*, pages 39–48. IEEE Computer Society, 2011.

[WHD+05] Hengqing Wen, Peter Yih-Ru Huang, John Dyer, Andy Archinal, and John Fagan. Countermeasures for GPS signal spoofing. In *Proc. 18th Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS)*, pages 1285–1290. Institute of Navigation, 2005.

[WJ03] Jon S. Warner and Roger G. Johnston. GPS spoofing countermeasures. *Homeland Security Journal*, 2003.

[WL09] Ting Wang and Ling Liu. Privacy-aware mobile services over road networks. In *Proc. 35th International Conference on Very Large Data Bases (PVLDB)*, pages 1042–1053, 2009.

[WRH12] Kyle Wesson, Mark Rothlisberger, and Todd Humphreys. Practical cryptographic civil GPS signal authentication. *Journal of The Institute of Navigation (NAVIGATION)*, 59(3):177–193, 2012.

[WXH+12] Yu Wang, Dingbang Xu, Xiao He, Chao Zhang, Fan Li, and Bin Xu. L2p2: Location-aware location privacy protection for location-based services. In *Proc. 31st Annual IEEE International Conference on Computer Communications (INFOCOM)*, pages 1996–2004. IEEE Computer Society, 2012.

[XC09] Toby Xu and Ying Cai. Feeling-based location privacy protection for location-based services. In *Proc. 16th ACM Conference on Computer and Communications Security (CCS)*, pages 348–357. ACM Press, 2009.

[XDZ09] Kexin Xie, Ke Deng, and Xiaofang Zhou. From trajectories to activities: a spatio-temporal join approach. In *Proc. 1st International Workshop on Location based social networks (LBSN)*, pages 25–32. ACM Press, 2009.

[XKP09] Mingqiang Xue, Panos Kalnis, and Hung Keng Pung. Location diversity: Enhanced privacy protection in location based services. In *Proc. 4th International Symposium on Location and Context Awareness (LoCA)*, volume 5561 of *Lecture Notes in Computer Science*, pages 70–87. Springer, 2009.

[XZLX10] Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. Finding similar users using category-based location history. In *Proc. 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS)*, pages 442–445. ACM Press, 2010.

[YCP+11] Zhixian Yan, Dipanjan Chakraborty, Christine Parent, Stefano Spaccapietra, and Karl Aberer. SeMiTri: a framework for semantic annotation of heterogeneous trajectories. In *Proc. 14th International Conference on Extending Database Technology (EDBT)*, pages 259–270. ACM Press, 2011.

[YCP+13] Zhixian Yan, Dipanjan Chakraborty, Christine Parent, Stefano Spaccapietra, and Karl Aberer. Semantic trajectories: Mobility data computation and annotation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(3):49, 2013.

[YJHL08] Man Lung Yiu, Christian S. Jensen, Xuegang Huang, and Hua Lu. Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In *Proc. 24th*

International conference on Data Engineering (ICDE), pages 366–375. IEEE Computer Society, 2008.

[Yu08] Shijun Yu. *Contextualised and personalised location-based services.* PhD thesis, École Polytechnique Fédérale De Lausanne, February 2008.

[ZWZ+08] Yu Zheng, Longhao Wang, Ruochi Zhang, Xing Xie, and Wei-Ying Ma. GeoLife: Managing and understanding your past life over maps. In *Proc. 9th International Conference on Mobile Data Management (MDM)*, pages 211–212. IEEE Computer Society, 2008.

Publications

[DPM] Xihui Chen, David Fonkwe, and Jun Pang. Post-hoc user traceability analysis in electronic toll pricing systems. In *Proc. 7th and 5th International Workshop on Data Privacy Management and Autonomous Spontaneous Security (DPM/SETOP)*, volume 7731 of *Lecture Notes in Computer Science*, pages 29–42. Springer, 2012.

[CCS] Xihui Chen, Carlo Harpes, Gabriele Lenzini, Miguel Martins, Sjouke Mauw, and Jun Pang. Demonstrating a trust framework for evaluating GNSS signal integrity. In *Proc. 20th ACM Conference on Computer and Communications Security (CCS)*, pages 1329–1332. ACM Press, 2013.

[ERCIM] Xihui Chen, Carlo Harpes, Gabriele Lenzini, Sjouke Mauw, and Jun Pang. Location assurance and privacy in GNSS navigation. *ERCIM News*, 2013(94), 2013.

[ECML/PKDD] Xihui Chen, Piotr Kordy, Ruipeng Lu, and Jun Pang. MinUS: Mining user similarity with trajectory patterns. In *Proc. 7th European Conference on Machining Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science. Springer, 2014. To appear.

[CSF] Xihui Chen, Gabriele Lenzini, Miguel Martins, Sjouke Mauw, and Jun Pang. A trust framework for evaluating GNSS signal integrity. In *Proc. 26th IEEE Security Foundations Symposium (CSF)*, pages 179–192. IEEE Computer Society, 2013.

[ARES] Xihui Chen, Gabriele Lenzini, Sjouke Mauw, and Jun Pang. A group signature based electronic toll pricing system. In *Proc. 7th International Conference on Availability, Reliability and Security (ARES)*, pages 85–93. IEEE Computer Society, 2012.

[JoWUA] Xihui Chen, Gabriele Lenzini, Sjouke Mauw, and Jun Pang. Design and formal analysis of a group signature based electronic toll pricing system. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, 4(1):55–75, 2013.

[APWeb] Xihui Chen, Ruipeng Lu, Xiaoxing Ma, and Jun Pang. Measuring user similarity with trajectory patterns: Principles and new metrics. In *Proc. 16th Asia-Pacific Web Conference (APWeb)*, Lecture Notes in Computer Science. Springer, 2014. To appear.

[CODASPY12] Xihui Chen and Jun Pang. Measuring query privacy in location-based services. In *Proc. 2nd ACM Conference on Data and Application Security and Privacy (CODASPY)*, pages 49–60. ACM Press, 2012.

[CODASPY13] Xihui Chen and Jun Pang. Exploring dependency for query privacy protection in location-based services. In *Proc. 3rd ACM Conference on Data and Application Security and Privacy (CODASPY)*, pages 37–48. ACM Press, 2013.

[GeoInformatica] Xihui Chen and Jun Pang. Protecting query privacy in location-based services. *GeoInformatica*, 18(1):95–133, 2014.

[SAC] Xihui Chen, Jun Pang, and Ran Xue. Constructing and comparing user mobility profiles for location-based services. In *Proc. 28th Annual ACM Symposium on Applied Computing (SAC)*, pages 261–266. ACM Press, 2013.

[TWEB] Xihui Chen, Jun Pang, and Ran Xue. Constructing and comparing user mobility profiles. *ACM Transactions on the Web (TWEB)*, 2014. To appear.

[ICFEM] Xihui Chen, Ton van Deursen, and Jun Pang. Improving automatic verification of security protocols with xor. In *Proc. 11th International Conference on Formal Engineering Methods (ICFEM)*, volume 5885 of *Lecture Notes in Computer Science*, pages 107–126. Springer, 2009.

[NAVITEC] Carlo Harpes, Miguel Martins, Xihui Chen, Gabriele Lenzini, Sjouke Mauw, and Jun Pang. Implementation and validation of a localisation assurance service provider. In *Proc. 6th ESA Workshop on Satellite Navigation Technologies (NAVITEC)*, pages 1–8. IEEE Computer Society, 2012.

Index of subjects

α-user specified innocent, 64

a priori preference, 29, 58
abduction, 16, 29
activity, 6, 85
activity privacy, 85, 113
activity trajectory, 86, 92, 114
adversary model, 23, 51, 88
algorithm, 33, 35, 65
anonymisation, 49, 88
anonymiser, 51
area generalisation, 49
attenuation, 3, 21
attribute, 37

Bayesian theorem, 52, 94
behaviour perturbation, 114
beyond suspicion, 63

C/A code, 13
causal relation, 17
Chi-square test, 103
civil signals, 13
cloaking, 49, 88
clock offset, 37
clustering, 66
combination of GNSS signals, 21
conditional opinion, 17, 30
conditional reasoning, 16
consensus, 17
context-aware, 2
context-aware privacy, 50
contextual information, 6, 114
continuous time, 87
correctness increase ratio, 72

de-anonymisation, 94
de-obfuscation, 97
decomposition, 96
deduction, 16
degradation function, 27
density estimation, 62, 103
dependency between users, 115
distance of measurements to reference sets, 27
Doppler ratio, 37
Doppler shift, 24
dynamic context, 54, 56, 114

efficiency, 43, 77
entropy, 64
estimation error, 89, 105
exposed trajectory, 87
exposure event, 87

gamma distribution, 103
generalisation, 6
generalisation algorithm, 51, 65
global navigation satellite systems, 3
global positioning system, 3
GNSS, 21
GNSS receiver, 21
GNSS signal spoofing, 3, 15, 23

height difference, 37
historical k-anonymity, 50
history window, 59, 74

integration, 98
integrity level, 36
integrity opinion, 26, 32
isb conditional opinion, 31

jamming, 3

k-anonymity, 49
k-approximate beyond suspicion, 64
k-means, 75
kernel smoothing, 62, 103
knowledge, 56, 88

LAP, 42
linkability, 88
localisation algorithm, 22

localising attack, 99
location assurance, 4, 113
location privacy, i, 4, 85
location semantics, 86
location-based service, 2
location-based services, i
log-likelihood, 102
LPPM, 85, 87

malware, 23
Markov chain, 57, 89
Markov chian, 101
maximum likelihood estimation, 102
measurement, 63
metric, 63, 89, 105
minimum transition time, 90
mobility profile, 89
Monte Carlo method, 98

natural factor, 21
navigation data, 13
navigation message authentication, 19
normal distribution, 27, 38

obfuscation, 6
observed request trace, 57
observed trajectory, 88
optimisation, 94, 98

P(Y) code, 13, 24
personalised information, 1
perturbation, 88
PoI, 86, 93
point of interest, 86
Poisson process, 93
possible innocence, 64
power correlation, 24
privacy, 113
privacy enhancing technologies, 5
private information retrieval, 80
probable innocence, 64
proposition, 16, 23
prototype, 36
pseudonym, 49, 88, 89
pseudorange, 14, 15

quasi-identifiers, 49
query, 2
query dependency, 56, 57, 114
query privacy, i, 5, 113

receiver, 14, 21
reference measurement, 24
reference set, 25
request history, 56
request issuing pattern, 89

satellite, 21
security requirement, 113
semantic trajectory, 86
signal attributes, 24
signal integrity, 5, 22, 28
signal originator, 22
signal strength, 24
signal-to-noise ratio, 28, 37
simulated annealing, 98
spatial distribution, 51
spatial generalisation, 51
spoofing detection method, 19, 24, 28
ssb conditional opinion, 31
stateful spoofing detection, 25, 29
stateless spoofing detection, 25, 28
static context, 54, 114
stay time, 91
subjective logic, 16, 26

temporal, 114
time interval between requests, 61
tracking attack, 85
trajectory, 87
transition time, 90
travel history, 89, 101
trust framework, 20, 113

uncertainty, 16, 26, 64
user dependency, 94
user profile, 114
user profiles, 54

valid measurement, 25
validity opinion, 26
VETO opinion, 35
vulnerability, 4

Curriculum Vitae

2010 – 2014	Ph.D. student in the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg.
2007 – 2010	Master of Computer Science, Shandong University, China.
2008 – 2009	Master of Computer Science, University of Luxembourg.
2005 – 2007	Undergraduate counsellor, School of Pharmaceutical Science, Shandong University, China.
2001 – 2005	Bachelor of Computer Science, Shandong University, China

Born on April 29, 1982, Shandong, China.