CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	A cryptographic approach to location privacy
Authors	Eshun, Samuel N.
Publication date	2023
Original Citation	Eshun, S. N. 2023. A cryptographic approach to location privacy. PhD Thesis, University College Cork.
Type of publication	Doctoral thesis
Rights	© 2023, Samuel Nana Eshun. - https://creativecommons.org/licenses/by/4.0/
Download date	2026-08-03 16:54:00
Item downloaded from	https://hdl.handle.net/10468/15931

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

A Cryptographic Approach to Location Privacy

Thesis presented by

Samuel Nana Eshun, MSc

0000-0001-6897-2373

for the degree of

Doctor of Philosophy

NATIONAL UNIVERSITY OF IRELAND, CORK

COLLEGE OF SCIENCE, ENGINEERING AND FOOD SCIENCE

SCHOOL OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

April 2023

Head of School: Professor Utz Roedig

Supervisor: Dr. Paolo Palmieri

Contents

List of Figures iv
List of Tables v
Acknowledgements viii
Abstract ix

1 Introduction **1**
1.1 Motivation and Background 1
1.2 Location-Based Services (LBS) 2
1.2.1 Applications of Location Based Services (LBS) 3
1.3 Privacy Concerns 4
1.3.1 Privacy Threats 6
1.3.1.1 Areas of Interest 6
1.3.1.2 Identification 6
1.3.1.3 Linkability 6
1.3.1.4 Content Disclosure 7
1.3.2 Legislation on Privacy 7
1.4 Structure of the Thesis 8
1.5 Contributions/Publications 9

2 Preliminaries **12**
2.1 Adversary Models 12
2.1.1 Semi-Honest Adversary 12
2.1.2 Malicious Adversary 13
2.2 Homomorphic Encryption (HE) 13
2.3 Oblivious Transfer (OT) 14
2.4 Garbled Circuit (GC) 15
2.5 Bloom Filter (BF) 16
2.6 The Private Hadamard Product 18
2.7 Spatial Bloom Filter (SBF) 18

3 Literature Review **20**
3.1 Overview 20
3.2 General Privacy-Preserving Protocols for LBS 21
3.2.1 The Use of Homomorphic Encryption Techniques 21
3.2.1.1 Partially Homomorphic Encryption 23
3.2.1.2 Using Fully Homomorphic Encryption 25
3.2.2 Solutions Based on Compact Data Structure Algorithms 26
3.2.3 Using Differential Privacy (DP) for LBS 27
3.2.4 The Use of Oblivious Transfer Protocol in Location-Based Services 28
3.2.5 Additional Proposed Solutions for Location-Based Privacy-Preserving Protocols 29
3.3 Privacy-Preserving Protocols for Indoor Environment 30
3.3.1 Homomorphic Encryption (HE) 30

3.3.2 Privacy-Preservation Based on Compact Data Structure Algorithms . . . 32
3.4 The use of Obfuscation Techniques in LBS . . . 33
3.5 Server-Aided Location Privacy-Preserving Protocols . . . 35
3.6 Overview of De-anonymization Techniques in Location Datasets 36
3.6.1 Location Dataset . . . 36
3.6.1.1 Sources of Location Data/Collection of Location Data . . . 37
3.6.2 Location De-anonymization . . . 37
3.6.2.1 Target Dataset . . . 37
3.6.2.2 Auxiliary Data/Information . . . 38
3.6.3 De-anonymization Techniques . . . 38
3.6.3.1 Mobility Models . . . 39
3.6.3.2 Areas of Interest (AoI) . . . 40
3.6.3.3 Random Locations/Spatio-Temporal Points . . 42

4 Efficient De-anonymization Attacks on Location Data Using Hidden Markov Model 44
4.1 Introduction . . . 44
4.1.1 Contribution . . . 47
4.2 Preliminaries . . . 47
4.2.1 The DBSCAN . . . 48
4.2.2 Hidden Markov Model (HMM) . . . 49
4.3 Hidden Markov Model (HMM) for Location Trajectories . . . 49
4.4 Experimental Setup and Pre-processing . . . 52
4.5 De-Anonymizers . . . 55
4.5.1 Likelihood . . . 57
4.5.2 The Forward De-anonymizer . . . 58
4.5.3 Kullback-Leibler Divergence De-anonymizer . . . 59
4.6 Experimental Evaluation . . . 60
4.6.1 Dataset . . . 61
4.6.2 Experimental Results . . . 62
4.6.3 Comparison to Previous Attacks . . . 66
4.7 Related Work . . . 67
4.8 Conclusion . . . 69

5 A Privacy-Preserving Protocol for Indoor Wi-Fi Localization 70
5.1 Introduction . . . 70
5.1.1 Contribution . . . 72
5.2 Private Indoor Wi-Fi Localization . . . 73
5.2.1 The Two-Party Protocol . . . 75
5.3 Security . . . 81
5.3.1 Security Discussion . . . 81
5.4 Conclusion . . . 82

6 A Cryptographic Protocol for Efficient Mutual Location Privacy Through Outsourcing in Indoor Wi-Fi Localization 84

6.1 Introduction 84
6.1.1 Related Work 88
6.1.2 Contribution 91
6.2 Private Indoor Wi-Fi Localization 93
6.2.1 Protocol Participants 94
6.2.2 Protocol Overview 95
6.2.2.1 Preparation and Initialization Phase 96
6.2.2.2 Measuring Phase 98
6.2.2.3 Computation Phase 99
6.2.2.4 Retrieval Phase 100
6.2.2.5 Server Localization Phase 100
6.3 Security 102
6.3.1 Privacy 103
6.3.2 Security Definition 105
6.3.3 Security Proof 107
6.3.4 Potential Expansion to a Malicious Service Provider . . . 109
6.4 Evaluation 110
6.4.1 Computational Complexity 111
6.4.2 Communication Complexity 113
6.4.3 Communication and Computation for SP and CS 114
6.5 Conclusion 116

7 Conclusions & Future Work **117**
7.1 Conclusions 117
7.1.1 De-anonymization Attacks on Real-World Location Dataset 118
7.1.2 A Privacy-Preserving Protocol for Indoor Wi-Fi Localization 119
7.1.3 A Cryptographic Protocol for Efficient Mutual Location Privacy Through Outsourcing in Indoor Wi-Fi Localization 120
7.2 Future Work 121

List of Figures

1.1 A Categorization of Applications for Location-Based Services . . 3

2.1 1-out-of-2 Oblivious Transfer Protocol 15
2.2 An Example of Bloom Filter for Set Membership Check 17
2.3 Spatial Bloom Filter: Private Set Intersection 19

3.1 Location Privacy Protocols: Indoor and Outdoor Techniques . . 22

4.1 Fully Connected Hidden Markov Model for Area of Interest . . . 51
4.2 Successful De-anonymization Attack Adversary Model 53
4.3 Spatial Clustering of Taxicab Traces in Rome 54
4.4 Spatial Representation of a User's Movements in the GeoLife Dataset 56
4.5 Performance Rate of Forward De-anonymizer 62
4.6 Performance Rate of D_{KL} De-anonymizer 63
4.7 Comparing De-anonymizers in the Geolife Dataset 64
4.8 Performance Rate of the De-anonymizers on the Taxicabs [BBL$^+$]. 65
4.9 GeoLife Dataset: Performance Comparison of Our Method and Others 68

5.1 Wi-Fi Localization System Architecture 74
5.2 Two-Party Computation System Design With Private Inputs . . . 76

6.1 Secure Three-Party Computation: System Architecture 95

List of Tables

4.1 Notation: Acronyms and Symbols 48
4.2 Performance Comparison: D_{KL} vs D_{FW} Model on GeoLife Dataset 65
4.3 Performance Comparison: D_{KL} vs D_{FW} Model on Taxicab Dataset 65

5.1 Notation: Acronyms and Symbols. 73
5.2 Wi-Fi Fingerprint Reference Database 75

6.1 Notation: Acronyms and Symbols. 92
6.2 Radiomap for Wi-Fi Fingerprint Reference Database 94
6.3 Comparison of Computational Complexity at Client-Side. . . . 111
6.4 Time Required by the User to Perform the Required Encryption Computation for Different Key Sizes. The Table Details the Mean Encryption Times in Milliseconds (ms) Averaged Over 10 Encryption Rounds. This Illustrates the Relationship Between Increasing Key Size and Encryption Duration, Highlighting the Computational Impact of Stronger Encryption in Secure Communications. 112
6.5 Comparison of Bandwidth Usage at Client-side 113
6.6 Computational and Communication Complexity at the SP and CS. 115

I, Samuel Nana Eshun, MSc, certify that this thesis is my own work and has not been submitted for another degree at University College Cork or elsewhere.

Samuel Nana Eshun, MSc

This success is dedicated to my amazing family: Alberta, who has been my rock throughout this road; Lauraine, who has been my shining star; and Samuel, who has been my inspiration.

Acknowledgements

First and foremost, I extend my sincerest gratitude to my supervisor, Paolo, whose role in my life transcended the boundaries of academia. Our genuine friendship has made this journey not only more meaningful but also more enjoyable. Paolo's unwavering support and guidance throughout my PhD studies have been invaluable. I feel truly blessed to have had him by my side. I want to express my gratitude to Utz, the department head, for his continuous support as I wrote and advanced with my thesis. His support has been invaluable in keeping me focused on my goals. I am beyond grateful to my loving wife, Alberta, who has been my rock and the source of my strength. Her unyielding support, understanding, and belief in my abilities have laid the foundation for my academic achievements. I am eternally thankful for her love and sacrifices, which have allowed me to reach this milestone. I dedicate this accomplishment to my beautiful daughter Lauraine, who has inspired me throughout this journey. Her presence has been a beacon of hope and motivation, pushing me to strive for excellence as a scholar and a father.

I am incredibly thankful to my mother, Sarah, for her steadfast belief in me and support throughout my career. My father, Moses, also deserves special recognition for his significant role in my life and the support provided by all my family members. I extend my deepest thanks to Ivan for his technical support and encouragement during the challenging moments of this journey. His expertise and willingness to help were instrumental in overcoming obstacles along the way. My heartfelt appreciation goes to all my colleagues in the security lab, particularly Prathyusha, Amir, Apurva, Buvana, Alia and Piotr. The camaraderie shared experiences and the good times together have created a truly memorable experience. I am grateful to have had the opportunity to work alongside such talented and supportive individuals. Though I cannot mention everyone by name, I cherish each of you. Lastly, I thank Michael for guiding me through my driving lessons in Ireland. His patience and guidance made a significant impact on my life. To all of you, I am eternally grateful. This accomplishment would not have been possible without your love, support, and faith in me. Thank you for being the pillars upon which I have built my dreams.

Cork, December 2022

Abstract

The rapid expansion of location-based services (LBS) has driven an escalating demand for personalised and context-aware applications, enriching user experiences across health, weather, and navigation sectors. These services offer valuable insights into various applications using large-scale datasets of individual mobility traces. However, alongside the benefits come considerable privacy concerns, as the data could inadvertently reveal sensitive information about users' movements and behaviours. This thesis delves into multiple facets of privacy within LBS, concentrating on the de-anonymisation of mobility data and the subsequent privacy risks it poses. In response to these concerns, the thesis presents privacy-preserving indoor localisation techniques, later improved with an efficient cryptographic protocol to protect users and service providers against privacy breaches. The first part of this thesis focuses on de-anonymisation attacks on mobility data. We propose a novel de-anonymisation model that employs hidden Markov models (HMM) to create user mobility profiles based on spatiotemporal trajectories. The performance of this model is assessed using real-world mobility datasets from two different cities, Shanghai and Rome. Our attack techniques significantly improve over existing de-anonymisation techniques, successfully re-identifying up to 85% and 90% of anonymised users in the respective datasets. However, despite the model's effectiveness, limitations exist, such as the model's dependence on the availability of a training dataset. Future work could explore unsupervised machine learning algorithms to address these limitations or utilise more sophisticated techniques like recurrent neural networks (RNNs) to model the evolution of user mobility behaviour over time.

The second part of the thesis addresses privacy concerns in indoor Wi-Fi localisation. We propose a privacy-preserving protocol that uses partial homomorphic encryption (Paillier's cryptosystem) to guarantee user location privacy while allowing computation in the encrypted domain. This approach ensures that most of the computational overhead on the user side is delegated to the server while hiding the user's exact location. By leveraging the Spatial Bloom filter data structure, complemented by homomorphic encryption, the service provider can learn about the user's presence in predefined areas without revealing the user's exact location or these predefined areas to the user.

The third part of the thesis introduces an efficient and privacy-preserving cryptographic protocol that incorporates a more realistic security assumption in the

form of a malicious adversary, one of the most important improvements to guarantee privacy for both the service provider and the user, unlike the *semi-honest adversary in the previous protocol*. Our protocol employs additive homomorphic encryption (DGK encryption) to preserve the privacy of the user's location fingerprint while allowing the service provider to compute over the encrypted fingerprint. In addition, garbled circuits protect the service provider's reference database against malicious users while delivering location output to the user. Finally, spatial Bloom filters further enhance the protocol by allowing the service provider to learn the user's vicinity in predefined areas of interest without revealing the exact location to the user or these predefined areas.

Compared to similar protocols, our proposed solution demonstrates a significant reduction in computational costs on the user side and a 99.99% reduction in online communication costs, making it more efficient and practicable in the Internet of Things environments. Furthermore, our protocol is the first to provide security against malicious users, whereas other protocols are limited to honest-but-curious adversaries.

For future work, we recommend strengthening the protection against actively corrupt service providers or cloud services by implementing additional cryptographic techniques, such as the ABY framework, for efficient mixed-protocol multiparty computation. Moreover, exploring other protocols or cryptographic primitives that improve efficiency, security, and privacy is encouraged, possibly through the combination of different techniques to optimise the protocol's current efficiency or reduce the size of the garbled circuit.

By examining the challenges posed by de-anonymisation attacks and developing innovative solutions, this thesis offers a comprehensive approach to enhancing privacy and security in location-based services. By investigating privacy-preserving indoor localisation techniques and developing efficient protocols, we strive to protect the interests of both users and service providers. As the demand for personalised and context-aware applications continues to grow, this research contributes significantly to the ongoing conversation surrounding privacy and data protection in the digital age.

The importance of addressing privacy concerns in location-based services cannot be overstated. As technological advancements progress and LBS permeate various aspects of our lives, ensuring the confidentiality of user data becomes paramount. The solutions presented in this thesis, including privacy-preserving indoor localisation techniques and efficient cryptographic protocols, are crucial

steps towards achieving a balance between the benefits offered by these services and the privacy requirements of users and service providers.

As location-based services evolve, new challenges and privacy risks will undoubtedly emerge. Therefore, the work presented in this thesis should be considered part of an ongoing effort to develop and refine techniques that preserve user privacy while maintaining the functionality and efficiency of LBS. The exploration of alternative cryptographic primitives, the improvement of existing protocols, and the development of new privacy-preserving methods will be essential to ensuring the continued growth and success of location-based services in a secure and privacy-conscious manner.

In conclusion, this thesis comprehensively examines privacy concerns in LBS, focusing on de-anonymising mobility data and developing privacy-preserving indoor localisation techniques. The efficient cryptographic protocols proposed to offer robust protection for both users and service providers, paving the way for a more secure and privacy-oriented future for LBS.

Chapter 1

Introduction

1.1 Motivation and Background

Location-based services (LBS) have seen an uptick in popularity due to the rising demand for mobile devices equipped with navigation systems. Generally, these LBS are applications that use location data to provide personalized and relevant services to users. Navigation services, weather and health are all examples of such services.

As more people get interested in and subscribe to these LBS-targeted services, service providers' drive for additional customers intensifies, resulting in an exhaustive collection of ever more valuable mobility datasets. Undoubtedly, using LBS has significant benefits; nevertheless, this also raises privacy concerns because these LBS-targeted services rely primarily on individual mobility traces, which are often unique.

The use of location datasets alone is not the problem. Instead, the primary concern is the uniqueness of the information collected and the variety of organizations, including government agencies, that collect and potentially misuse these data.

The growing use of this LBS by governments to follow their populations in an effort to contain the spread of Covid-19 is a good example. Sometimes these datasets are exploited and even sold to third parties, including government agencies, without the consent of the people involved. In some jurisdictions, it's even possible for law enforcement agencies to obtain a mobility dataset without a warrant if they so choose. Ordinarily, this shouldn't be a problem until they

begin to build a profile of the behaviour of users that might subsequently be used against them.

In particular, the exploration of sensitive information (home, place of work, place of worship, etc.) from this mobility dataset should be a source of concern for many people.

As a result, there is a need to quantify the degree of a privacy breach to adequately safeguard the data collected through anonymization or, better yet, render the required LBS services in using privacy-preserving applications in order not to infringe on the users' privacy.

For example, it should be possible to provide weather forecast services to users without tracking them every second or revealing their precise location. Furthermore, it should be possible to track users in infrastructure locations of interest without knowing their exact location, and nothing should be known about their whereabouts when they are not in these locations of interest. Most importantly, individuals must be aware that their data is being collected, and they should have a say in whether the data should be collected or not, as well as what could be done with the data, including whether it can be given to third parties.

1.2 Location-Based Services (LBS)

The proliferation of smart devices equipped with positioning sensors has increased the use of location-based services (LBS). Location-based services, sometimes called location-aware services, utilize the location information of a mobile device to provide tailored services to end users. Real-time traffic updates, nearby points of interest (e.g. nearest pharmacy), weather information, and emergency call routing are just a few of the advantages of using LBS. One of the most popular applications of LBS is Google Maps, with over one billion monthly active users. End-users using the LBS are required to submit their location before receiving the required tailored services. However, users do not have to manually enter their location in order to access LBS because the availability of location services in these smart devices ensures that this is done automatically. This enhances the user experience, hence increasing the popularity and utility of the LBS. The Global Navigation System (GPS,GLONASS, Galileo), Cell ID (base transceiver station), and proximity positioning systems such as Wi-Fi access points can all be used to determine the location of a mobile device. However, the selection of technology in computing the location is gen-

erally influenced by a variety of factors such as availability and the surrounding environment (indoor and outdoor).

1.2.1 Applications of Location Based Services (LBS)

Location-based services (LBS) encompass a wide range of applications, and the following is a pictorial summary of the numerous categories these applications fall into. Figure 1.1 gives a graphical overview of these applications:

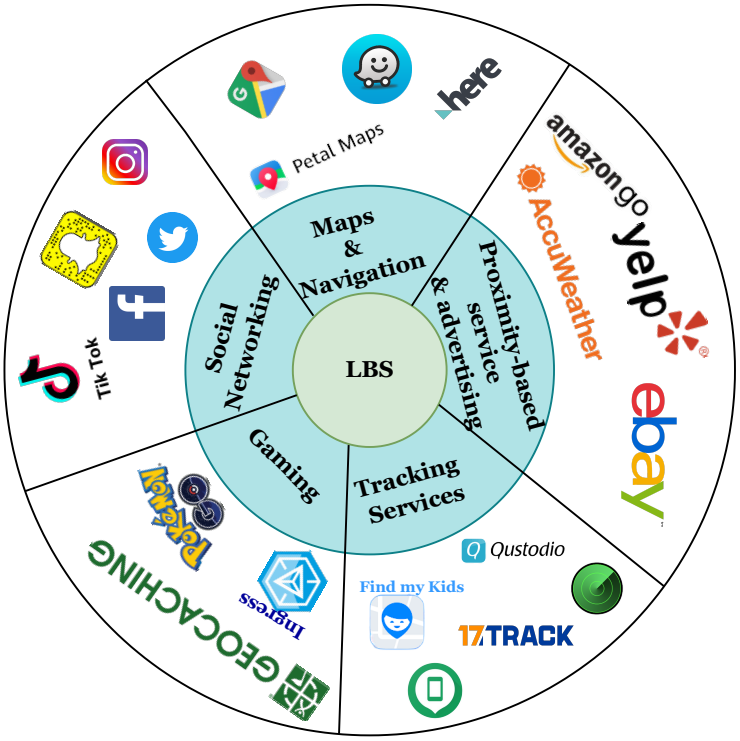


Figure 1.1: A Categorization of Applications for Location-Based Services Based on the Function they Serve. Each Slice Represents a Different Form of Location-Based Service that Users Subscribe to, Ranging from Navigation Services to Weather Forecasting.

Figure 1.1 depicts a comprehensive classification of various applications that utilise location-based services (LBS), which is categorised into distinct sectors based on their primary functions with each representing a different application category:

Maps & Navigation: This sector provides users with real-time directions, route planning, and location findings. Thus, these applications help users to find unfamiliar areas and even sometimes avoid traffic congestions routes. Popular examples include Google Maps, Waze, and HERE.

Proximity-based Services & Advertising: Applications in this category leverage on users' location to provide personalized and relevant information to users. For instance, users may receive shopping coupons they are nearby or recommendations about nearby restaurants, events, or weather updates. Examples of such applications include Yelp, eBay and AccuWeather.

Tracking Services: This type of applications enables tracking of assets, vehicles, and peoples (as a parental control app). Key features include geo-fencing, alert, and real-time location monitoring. Some of the known app are 'Find my Kids', 'Find My Device', Qustodio parental control and '17TRACK'.

Gaming: Applications in this section incorporate users' real location as part of the game while exploring their surroundings and they can also engage with other users within their vicinity. Examples include Pokémon Go, Ingress, Geocaching.

Social Networking: Location-based features in social networks like Facebook, Instagram, Tik Tok, Twitter, Snapchat and Foursquare allow users to share their location with friends, discover location-based content or nearby places and to connect with people around them.

1.3 Privacy Concerns

As location-based services gain in popularity and variety, service providers' efforts to attract new clients are becoming more aggressive. This has resulted in an enormous amount of location data being collected. The fact that the majority of the digital services we use in our everyday lives are equipped with location-based services should be a source of concern for many since the service providers are collecting location data at a finer granularity with each passing day. According to the New York Times (NYT), for example, over a billion data points from approximately 12 million phones in the United States (New York, Los Angeles, San Francisco and Washington) were discovered in an investigation that tracked individual movements, such as visits to psychiatric facilities and to even whom they spend the night with [1].

In some cases, these service providers may sell their customers' location data without their knowledge or permission. And even when consumers are notified, the default option is opt-out rather than opt-in consent, knowing full well

[1] www.nytimes.com

that the vast majority of people will not bother to read the information or even get as far as opting out. Life360, for example, makes millions of dollars by selling the precise location data of its users of the well-known family safety app, including the location history of children [2]. As a result, when this information falls into the hands of predators, it may be extremely perilous to the children concerned. Additionally, data publishers often believe that deleting personal identifiers from a dataset renders it anonymous and ready for publication. One such example is the Netflix Prize dataset. The Netflix prize draw, which was introduced in 2006, offered a substantial prize to anyone who could enhance Netflix's personalised movie recommendations by a 10% margin. The prize draw caught the attention of privacy researchers like Arvind Narayanan and Vitaly Shmatikov, who use auxiliary information to re-identify users in the Netflix Prize dataset [3]. Their method makes use of the de-anonymization technique on the Netflix Prize dataset, which contains the Netflix movie ratings of approximately 500 000 users. By using the Internet Movie Database (IMDb) as auxiliary information (a source of secondary information or to build background knowledge), the researchers were able to successfully de-anonymize the subscribed users of Netflix to uncover sensitive information like their religious beliefs and political preferences [NS08].

According to a publication that was released not too long ago by the BBC, a British company that is in the business of selling people's location data has admitted to getting this information without the agreement of the users who were involved [4]. This demonstrates how some of these companies are progressively abusing the data that is provided by their subscribers. Some of the data that were acquired provide a high level of granular details, which, when combined with people's location histories, can lead to the discovery of their true identities. The fact that users download apps for a particular purpose, yet the apps end up exchanging information for a purpose that is completely unrelated to the purpose for which the users downloaded the apps is another disturbing discovery. A new article published by Reuters demonstrates how the mobile app for the Canadian coffee business Tim Hortons continues to track and record users' location information even after the app is closed, which is in direct contravention of the privacy regulations that govern the country [5]. A two-year investigation discovered that the coffee company Tim Hortons had been collecting a massive

[2] www.themarkup.org
[3] https://www.forbes.com
[4] https://www.bbc.com/
[5] https://www.reuters.com/

amount of location data, including extremely sensitive information about its customers. When you look at all of these examples together, it's clearer than ever how often small and large organisations routinely breach the privacy of their most vulnerable users.

1.3.1 Privacy Threats

Even though LBS has several advantages, many people are sometimes ignorant of the possible consequences of using the technology. These concerns go beyond mere tracking and involve the gathering of background information. Some of these threats are classified as follows:

1.3.1.1 Areas of Interest

Areas of interest are places that people frequently visit and spend a significant amount of time in, including their homes, workplace, shopping mall, or theatre. In addition, certain areas of interest may be more sensitive to the user in question; for example, a health facility may reveal a person's health status or political party inclination. Occasionally, these areas of interest can lead to the individual's identity being exposed through the usage of auxiliary information.

1.3.1.2 Identification

This privacy threat allows an adversary to re-identify anonymous individuals using their Spatio-temporal mobility trajectories. For example, when an anonymised (personal identifiers removed) mobility dataset is made public, an attacker can mine this data and cross-reference it with secondary data in order to re-identify the individuals involved. Even in the absence of secondary information, it is sometimes possible to re-identify the target individuals, depending on how the personal identifiers are removed or by simply merging quasi-identifiers.

1.3.1.3 Linkability

The following type of privacy threat involves cross-referencing two or more accounts belonging to the same user, either anonymous or under different usernames. Specifically, the threats revolve around disclosing a relationship between different accounts in order to gain access to sensitive information that would not normally be available.

1.3.1.4 Content Disclosure

The threat of content disclosure refers to the possibility of revealing previously unknown information about a target user. Home addresses, email addresses, and telephone numbers are just a few examples of what is considered "content disclosure." Sometimes it is possible that a content disclosure may also lead to re-identification.

1.3.2 Legislation on Privacy

As specified in Article 12 of the 1948 Universal Declaration of Human Rights (UDHR) and as enshrined in the constitutions of numerous jurisdictions worldwide, privacy is a fundamental human right that must be safeguarded.

Despite the fact that many jurisdictions do not have particular legislation governing actions concerning location privacy, there are pockets of regulations in many jurisdictions that location privacy is occasionally subject to. Given that location data is considered personal data (i.e., personally identifiable information of a natural person), it is subject to many existing privacy rules.

In the United States of America (USA), Section 5 of the Federal Trade Commission (FTC) Act [6] outlaws deceit and unfair information practices in cyberspace. In addition, the Children's Online Privacy Protection Act (COPPA) [7] of 1998 is another law that appears to protect children's information flow, including location data and has been in effect since 1998. This act prohibits collecting information about children under the age of thirteen years in any form whatsoever. Other noteworthy privacy legislation in the United States, both at the state and federal levels, include the California Consumer Privacy Act [8](CCPA), which became effective in 2020, and the Health Insurance Portability and Accountability Act [9](HIPPA), which was passed in 1996.

The European Union, on the other hand, has comprehensive regulation regarding location data, dubbed the General Data Protection Regulation (GDPR) [10], which was published in 2016 and became effective in May 2018. The GDPR is a piece of legislation that applies to all EU member states. Articles 5 and 6, for example, establish the fundamental principles that underpin the legisla-

[6]https://www.ftc.gov/
[7]https://www.ftc.gov/
[8]https://oag.ca.gov/home
[9]https://www.cdc.gov/
[10]www.european-union.europa.eu/index$_en$

tion. These fundamental principles govern the processing of personal data as well as the lawfulness of the processing. It primarily focuses on lawfulness, fairness, transparency, purpose limitation, data minimization, accuracy, storage limitation, integrity and confidentiality, and accountability.

1.4 Structure of the Thesis

The rest of the thesis is structured as follows:

Chapter 2 outlines some of the cryptographic techniques and primitives utilised in this thesis to help the reader comprehend this thesis. We begin by explaining some of the types of adversary models that can corrupt a cryptographic protocol before moving on to some of the cryptographic primitives that are critical to understanding this dissertation.

Chapter 3 provides an overview of the literature most relevant to this thesis. In particular, the most recent and advanced attacks on location privacy are discussed, along with the effectiveness of the proposed techniques, including the splitting and selection criteria of the dataset used in these attacks. In addition, a discussion on some of the state-of-the-art solutions for privacy-preserving protocols in the context of addressing risks associated with location privacy is presented.

Chapter 4 details the proposed two novel and highly effective de-anonymization attacks on the location dataset and discuss them in length. The chapter also presents an in-depth performance analysis of the proposed techniques. In addition, a comprehensive performance analysis comparing the effectiveness of the proposed techniques in the thesis with those from the literature is further explored.

Chapter 5 presents solutions to privacy leakage in the context of the location dataset. In particular, the chapter presents a proposed location privacy-preserving protocol that protects not just the user's privacy but also the service provider's privacy while delivering the required service. The chapter further discusses the security robustness of the protocol in the context of honest-but-curious adversaries (Semi-honest adversaries) settings.

Chapter 6 expands on the work of Chapter 5 by proving a stronger security assumption in contrast to the somewhat lax security assumption in a heterogeneous Internet of Things (IoT) environment presented in the previous chapter.

The chapter improves security by providing protection against an active user adversary while also guaranteeing location privacy for both the user and the service provider. In addition, the protocol described in the preceding chapter is enhanced in this chapter by delegating the heavy-side computation performed by the user to a third party.

Chapter 7 concludes the thesis by highlighting some of the importance of location privacy in our everyday lives and the need to design privacy-preserving protocols to mitigate such risks. The chapter further discusses some of the potential future works that will be founded on this thesis.

1.5 Contributions/Publications

This thesis makes a number of contributions to the existing body of research. These contributions have been published and submitted to top journals, international conferences, workshops, and poster sessions. Specific contributions in each of the works submitted have been described in detail in the various chapters. The following lists of publications can be found below:

- Samuel N. Eshun and Paolo Palmieri. 2019. A privacy-preserving protocol for indoor Wi-Fi localization. In Proceedings of the 16th ACM International Conference on Computing Frontiers (CF '19)-Mal-IoT. Association for Computing Machinery, New York, NY, USA, 380–385.

- Samuel N. Eshun and Paolo Palmieri. Privacy-preserving protocol for indoor Wi-Fi localization. A poster presentation at the CySep summer school, co-located with the IEEE Symposium on Security and Privacy in Stockholm, Sweden, in 2019.

  In this work, we proposed a novel scheme that utilizes uses probabilistic data structure, specifically the Spatial Bloom Filter, to preserve the privacy of both users (requesters of location-based services) and service providers. By leveraging the additive homomorphic encryption algorithm, the scheme allows a service provider to privately have access to users' locations in real time without breaching the user's privacy while preventing data exposure. Our method is designed to work seamlessly with existing Wi-Fi infrastructures, providing a cost-effective and privacy-focused solution for indoor localization without requiring additional hardware. This harmonious integration underscores our commitment to enhancing pri-

vacy without compromising efficiency and accessibility.

- Samuel N. Eshun and Paolo Palmieri, "Two de-anonymization attacks on real-world location data based on a hidden Markov model," 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), 2022, pp. 01-09.

  In this published work, we introduced two novel hidden Markov based attacks that take into account both the temporal and spatial resolutions of the user mobility trajectories. In particular, our model employs a clustering algorithm to initially cluster users' areas of interest, before constructing user profiles based on the HMM. Our attack utilizes two algorithms: the Forward algorithm and a Kullback-Leibler divergence to perform the de-anonymization. Preliminary result based on real-world mobility dataset demonstrate an impressive performance, exceeding 80% in de-anonymization accuracy. This marks a significant improvement over existing benchmarks in the literature, underlining the robustness and precision of our approach.

- Samuel N. Eshun and P. Palmieri, "A cryptographic protocol for efficient mutual location privacy through outsourcing in indoor Wi-Fi localization," in IEEE Transactions on Information Forensics and Security, doi: 10.1109/TIFS.2024.3372805. This has been accepted for publication in IEEE Transactions on Information Forensics and Security, and is available under the "Early Access" area on IEEE Xplore [EP24].

  This work presents a significant advancement over the previously established two-party privacy-preserving protocol for indoor Wi-Fi localization. The enhanced protocol efficiently combines cryptographic primitives such as additive homomorphic encryption, garbled circuits, and spatial Bloom filters to protect the privacy of both service providers and users. Unlike the previous protocol that only assumes honest-but-curios users, this efficient protocol protects the SP's database against malicious adversaries and significantly reduces the computation cost and online communication.

- Samuel N. Eshun and Paolo Palmieri, "Two de-anonymization attacks on real-world location data based on a hidden Markov model. Submitted to Journal of Location Based Services.

  Building on preliminary findings presented at the 4th Location Privacy Workshop during the 2022 IEEE European Symposium on Security and

Privacy, this work introduces two novel de-anonymization methods rooted in the Hidden Markov Model, encompassing both the temporal and spatial influences of user mobility trajectories. We evaluated the performance of our attack algorithms;‘ the Forward algorithm and the Kullback-Leibler divergence using two publicly available real-world mobility traces from Shanghai and Rome. The results were remarkable, with our attack techniques successfully identifying up to 88% and 90% of anonymized users in the GeoLife and Taxicab datasets, respectively. This success rate significantly surpasses existing de-anonymization techniques, marking a substantial advancement in the field.

Chapter 2

Preliminaries

This section introduces the cryptographic tools and primitives that will be used throughout the remainder of the thesis to aid in understanding the rest of the thesis. In particular, we present the main techniques that served as the foundation for the development of this thesis.

2.1 Adversary Models

Formulating a security model for a cryptographic scheme is primarily determined by the adversary's capabilities. As a result, the security model provides insight into the security state of the scheme, as well as to what extent the adversaries are allowed to corrupt such a system. For the purposes of this thesis, only semi-*honest* and malicious adversaries are considered; for formal definitions, the reader is referred to [Gol04].

2.1.1 Semi-Honest Adversary

This type of adversary, also known as a passive adversary, follows the protocol's instructions in these settings. The corrupt party (adversary) attempts to learn more information than what the protocol permits. This implies that the agreed input/output of the function cannot be modified. As a result, any departures from the protocol's instructions are not allowed. This type of adversary is prevalent in privacy-preserving enhancing technologies.

2.1.2 Malicious Adversary

This type of adversary is also called an active adversary. When compared to semi-honest adversaries, protection against active adversaries is more expensive. In this type of setting, the corrupted party (adversary) can deviate from the protocol's instructions. Aside from choosing to ignore or make arbitrary inputs into the protocol's function, the adversary may also choose to issue an early abort in an attempt to gain an advantage.

2.2 Homomorphic Encryption (HE)

Homomorphic encryption techniques are semantically secure (INDistinguishability under Chosen Plaintext Attack, *IND-CPA*) cryptographic algorithms that enable the execution of particular mathematical operations on encrypted data. The protocol proposed in this thesis, in particular, makes use of the additive property of homomorphic encryption, which allows for the addition of encrypted data as well as the multiplication of ciphertext with a plaintext constant. For given plaintexts a, c and d, encryption and decryption function $\mathsf{Enc}()$ and $\mathsf{Dec}()$, and operations $\boxplus$ and $\boxdot$ for homomorphic addition and multiplication by a plaintext, respectively, we have:

$$\mathsf{Enc}\left(c+d\right) = \left(\mathsf{Enc}\left(c\right) \boxplus \mathsf{Enc}\left(d\right)\right) \quad , \tag{2.1a}$$

$$\mathsf{Enc}\left(a \cdot c\right) = \left(a \boxdot \mathsf{Enc}\left(c\right)\right) \tag{2.1b}$$

where the operations $(+, \cdot)$ are computed in the plaintext space, and $(\boxplus, \boxdot)$ in the ciphertext space.

Paillier encryption [Pai99] is a widely used additive homomorphic encryption scheme, and its security relies on the decisional composite residuosity problem. It has a public key size of T-bits RSA modulus $N = p \cdot q$, where p and q are two large primes. To encrypt a message x, let $\mathbb{Z}_N$ be a set of integers modulus N and $\mathbb{Z}_N^* \subseteq \mathbb{Z}_N$. Choose $g \in \mathbb{Z}_{N^2}^*$, the generator of the multiplicative group and a random number $r \in \mathbb{Z}_N^*$, then for $x \in \mathbb{Z}_N$ gives:

$$\mathsf{Enc}(x) = g^x r^N \mod N^2 \ , \tag{2.2}$$

and the public key being (g, N). The reader is referred to [Pai99] for a detailed description and the security proof of the scheme.

To improve efficiency over Paillier-based schemes, the protocol presented in this thesis employs the DGK [DGK07, DGK09] encryption technique, which has a smaller ciphertext compared to Paillier due to the lower exponent of the modulus. Without any modification, DGK works only on small message space (μ), because it was initially designed for secure comparisons of integers. The DGK hardness is based on the strong assumption of the RSA subgroup, and it performs better than the other additive homomorphic encryption schemes for both encryption and decryption. Compared to Paillier, the ciphertext space is T-bits of the RSA modulus $N = p \cdot q$ while Paillier is $2T$. Similar to other cryptographic algorithms, the scheme has three main parts:

Key Generation The process of generating a key starts by first choosing RSA modulus $N = p \cdot q$, where p and q are two large primes and $\log_2 p \approx \log_2 q$. Select a plaintext space μ of size $l - bits$ and another two distinct primes of size $t - bits$; p_t and q_t such that $p_t | p - 1$ and $q_t | q - 1$. Choose $g, h \in \mathbb{Z}_N^*$ the generators of $\mathbb{Z}_N^*$, such that g and h have orders $\mu p_t q_t$ and $p_t q_t$ respectively. Therefore, the public key is (N, g, h, μ) and the secret key (p, q, p_t, q_t).

Encryption To encrypt a message $x \in \mathbb{Z}_\mu$, choose $r \in \mathbb{Z}_N$ and compute:

$$\mathsf{Enc}(x) = g^x h^r \mod N \tag{2.3}$$

Decryption The decryption is simply raising the ciphertext $\mathsf{Enc}(x)$ to a power $p_t q_t$ modulus N, i.e.,

$$x = \mathsf{Enc}(x)^{p_t q_t} \mod N \tag{2.4}$$

Due to the small message space μ, it is possible to build a table of $g^{p_t q_t}$ values to optimize the decryption process.

Though DGK works on small message space μ, it can be modified to operate on large plaintext space by utilizing the Pohlig-Helman algorithm [PH78] to perform the decryption.

2.3 Oblivious Transfer (OT)

Oblivious Transfer (OT) is a two-party protocol where one party (the *sender*) holds a secret input of n bits or strings of bits, and the other party (the *receiver*) wants to obtain one of these bits/strings. The receiver wants the sender to remain oblivious to which bit she chooses, while the sender wants to ensure

only one bit is revealed to the receiver. When $n = 2$, we call it *1-out-of-2* OT. The protocol was first introduced in 1981 by Rabin [Rab81] and has since become a fundamental component in several cryptographic constructions [Bea95, NP05, NP01]. In detail, for a 1-out-of-2 OT protocol, a sender inputs m-pairs of bits $\{\mathcal{K}_i^0, \mathcal{K}_i^1\}$ and the receiver inputs her choice in the form of m-bits, $b_i \in \{0, 1\}$. At the end of the protocol, the receiver obtains $\mathcal{K}_i^{b_i}$ and learns no information about $\mathcal{K}_i^{1-b_i}$. The sender on the other hand obtains nothing and learns nothing about the receiver's choice b_i. This is illustrated in Figure 2.1, where both the sender (Alice) and the receiver (Bob) can jointly compute a function without revealing their respective inputs or outputs to one another.

Ongoing research, including [Bea95, NP01], aims to improve OT's effective implementation both in the semi-honest and malicious model. These improvements have led to significant gains, such that most of the expensive computation can now be shifted to the set-up phase.

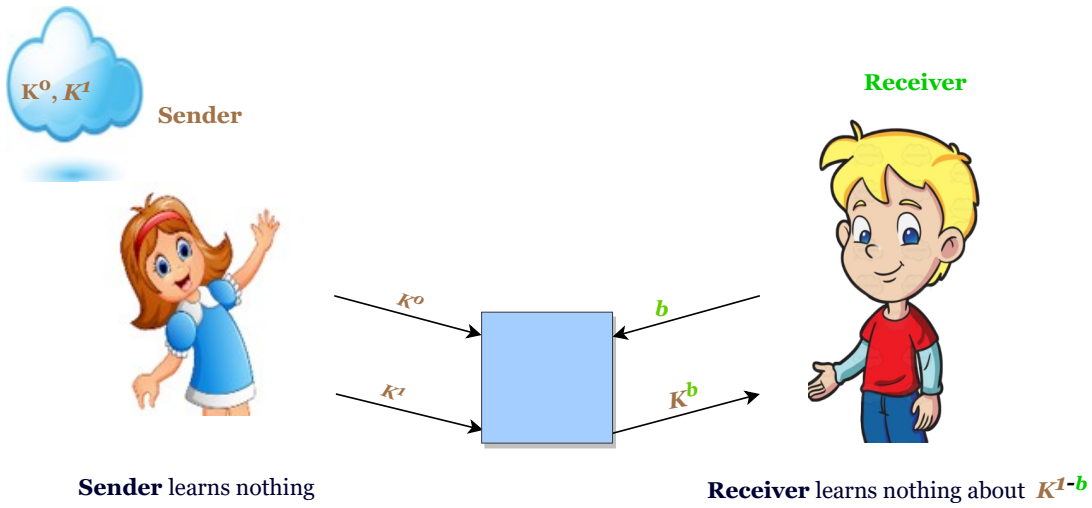


Figure 2.1: Illustration of 1-Out-of-2 Oblivious Transfer Protocol, Where the Sender (*Alice*) Inputs Two Keys (K^0 and K^1). The Receiver (Bob) Obtains Only One (K^b) of These Keys Based on His Choice Input Bit, (b) and Learns Nothing About the Other Key, i.e., K^{1-b}. The Sender (Alice) Also Learns Nothing About K^b.

2.4 Garbled Circuit (GC)

Garbled circuits [Yao82, Yao86], a work of Yao in the late 1980s, allow two parties to compute a function on their private input.

In detail, for a given Boolean circuit C, one party, called *generator* (S), con-

structs a garbled circuit G by assigning two random values/cryptographic keys $(\tilde{k}_i^0, \tilde{k}_i^1)$ for each wire w_i of the circuit. For each gate G_i, S encrypts using the generated random values to obtain an encrypted table (T_i) with each entry containing the output value's encryption. S permutes the table T_i's order and sends to the other party (called *evaluator*), the garbled circuits together with the garble values of the generator's input.

To evaluate the garbled circuits, the evaluator (E) and S engage in 1-out-of-2 OT to obtain the garbled values corresponding to the evaluator's plain inputs from S. Having received the garbled values for both parties, the evaluator can now evaluate the garbled circuit gate by gate to retrieve the output values from the output wires. Because the table is permuted, the output wires reveal nothing about the circuits' internal wires; thus, the evaluator learns only the output values and nothing else. Given a translation table, the output values are converted to their plain output values by the output party. We refer to [LP09] for the proof of security of Yao's protocol in the semi-honest model.

Several optimisation techniques have been proposed to improve the security of the original protocol. Some of these techniques include free XOR [KS08], point-and-permute [BMR90, MNPS04] and garbled row reduction [NPS99]. Further work by [PSSW09] needed only two ciphertext for AND gates but was not compatible with free XOR; this was later improved by [ZRE15], through a technique called half-gate, requiring only two ciphertexts for AND gates and zero-ciphertext for XOR gates (free XOR). All these techniques together have improved the efficiency of the protocol by reducing the number of non-XOR gates. Hence, they significantly reduce the communication and computation overhead for constructing and evaluating the circuit.

2.5 Bloom Filter (BF)

A Bloom filter [Blo70] is a probabilistic data structure used in representing a set of elements, which supports set-membership queries. The data structure is space-efficient, and elements are inserted into the filter using hash functions. When querying the set for membership of an element, a positive output from the BF implies the element belongs to the set minus a false positive probability. When the output for an element is negative then surely the member does not belong to the set, as the data structure is not subject to false negatives.

In particular, let $S = \{x_1, \cdots, x_n\}$ be a set of n-elements and $\mathcal{H} = \{h_1 \cdots, h_k\}$

be set of independent cryptographic hash functions. BF has an array of size m-bits that represent members of the set S. Before members of the set are inserted, the filter is initially set to zero. To insert a member x, for every $h \in \mathcal{H}$ compute:

$$h_j(x) \mapsto a \tag{2.5a}$$

$$B\{a\} = 1 \tag{2.5b}$$

where $a \in \{1, \cdots, m\}$ i.e., the range of positions from 1 to m on the filter (B).

To check if an arbitrary element $y \in S$, compute:

$$B[h_j(y)] = 1 \tag{2.6}$$

for all k hash functions. If any of the hash functions $h \in \mathcal{H}$ maps to 0, then the filter (B) correctly outputs that $y \notin S$. However, if all the hash functions $\mathcal{H}$ map to 1, then $y \in S$ minus the probability of a false positive. A false positive may arise because there are instances where $y \notin S$ but all the bit positions of y may be set to 1 due to the likelihood of collisions of the hash functions. The false-positive probability can be calculated for any given filter, as a function of the filter.

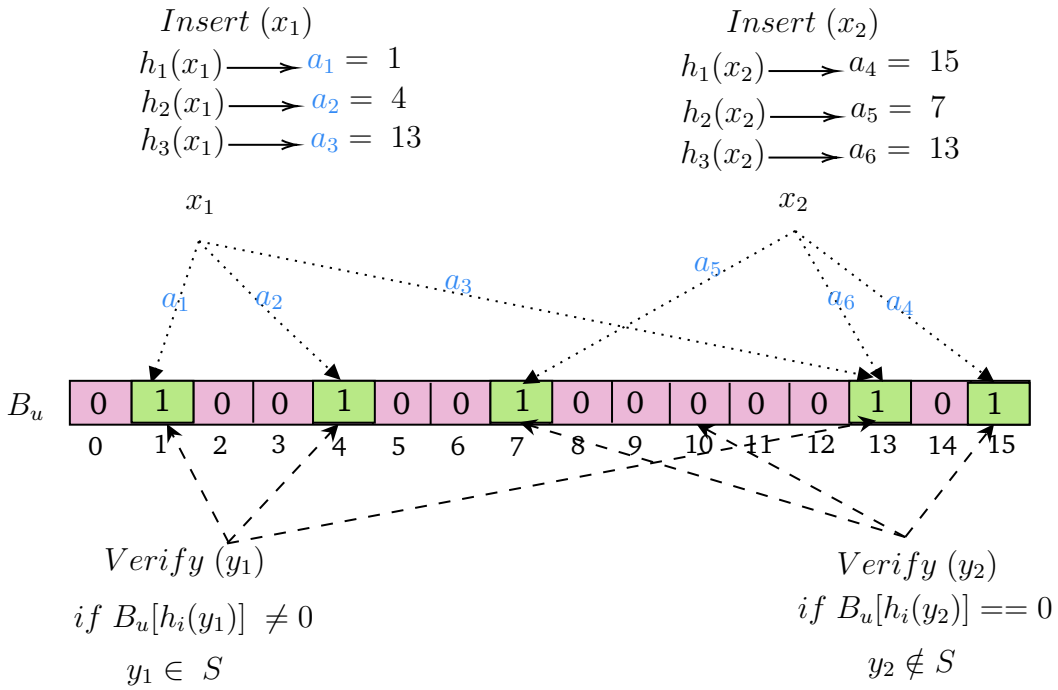


Figure 2.2: **An Example of a Bloom Filter B_u, of Size $m = 16$). Elements $x_1, x_2 \in S$ Are Inserted into the Filter B, by First Passing Them Through a Hash Function H, k Times, Where $k = 3$ in This Example. The Filter, Which Is First Initialized to 0s, Is Then Set at Bit 1 Based on the Indices (a_1) of the k Hashes (Modulo m). An Element y Is Verified by Hashing It k Times Modulo m, and If Any of the Index (a_i) Is Set to 0 on the Filter, Then $y \notin S$.**

2.6 The Private Hadamard Product

The Hadamard product, also known as entry-wise or elementwise multiplication, is essentially an operation that is performed on two matrices of the same size. Therefore, assume that A and B are two matrices (vectors for this thesis) with the same dimension, m x n. Then the Hadamard product, $\boxdot$, can be defined as $[A \boxdot B]_{ij} = [A_{ij}][B_{ij}]$, $\forall\ 1 \leq i \leq m$, and $1 \leq j \leq n$ [Mil07]. Similarly, in this project, multiplication involves two vectors of the same dimension, one being binary and the other involving elements of natural numbers, $\mathbb{N}$. However, the computation involving the Hadamard product in this work is done privately using a multiplicative homomorphic algorithm. This is because these vectors are in the encrypted domain, in which case only the party with the correct decryption key can see the correct output; hence, the name "private Hadamard product."

2.7 Spatial Bloom Filter (SBF)

Spatial Bloom filters (SBF) [PCM14] are an extension of the original Bloom filter, which can also be used for set membership queries in a space-efficient manner. Unlike the original Bloom filter, SBF is a byte array and allows multiple sets to be stored in a single filter. It was originally constructed for spatial representation, where a predefined area is considered as one set. Two kinds of false positives can arise in querying an SBF: similarly to the BF, a false positive can occur when an element is wrongfully assumed to belong to a set; additionally, a false positive can also occur when an element is recognised as belonging to a particular set of those included in the filter, while in fact it belongs to another (this false positive case, which is specific to the SBF, is known as an *inter-set error*). Similar to BF, false negatives are not allowed. In an SBF, each set is represented by an integer numerical value, and such values are written into the filter during construction instead of the 1s that would be written in a

classical BF. We refer to [CPM15, CPM18] for a detailed analysis of the filter and its probabilistic properties. Figure 2 shows an example of a spatial Bloom filter $\mathsf{Enc}(B_u^{\#})$; which is a product of Bloom filter (B_u) and encrypted spatial Bloom filter $\mathsf{Enc}(B^{\#})$. The encrypted filter simply means the values encoded in the filter are not in the plaintext as shown in the diagram.

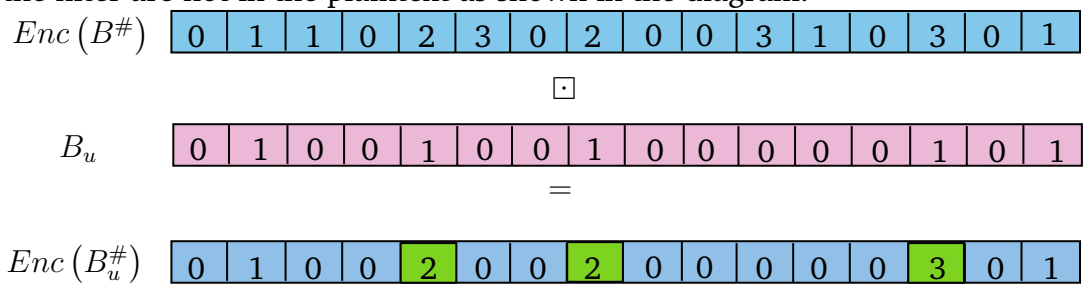


Figure 2.3: **Example of Spatial Bloom Filter: The Private Hadamard Product of $\mathsf{Enc}(B^{\#})$ and B_u, Producing SBF $\mathsf{Enc}(B_u^{\#})$. For Clarity and Ease of Reading, the Image Depicts Plaintext Values (Rather Than the Corresponding Ciphertexts) for $\mathsf{Enc}(B^{\#})$ and $\mathsf{Enc}(B_u^{\#})$ (in Light Blue).**

Chapter 3

Literature Review

3.1 Overview

When privacy risks are not handled well, they create security challenges for not just honest users but also for service providers, who are also at risk from corrupt users. As described in chapter 4 of this dissertation, privacy risk is not limited to the protocols themselves but also to data acquired on users because of how the data collected (which has a high degree of uniqueness of mobility traces) might be exploited or because the data is not stored securely.
In light of this, numerous solutions utilizing a variety of cryptographic techniques have been presented in the literature to preserve the privacy of users. These approaches include homomorphic encryption, secret sharing, secure multi-party computing, and others.
Given the fact that there is no one-size-fits-all solution for privacy-preserving-based services protocols, each protocol has its own set of advantages and disadvantages for any given application. Some of these methods are aimed at protecting user query privacy while also preventing tracking, profiling (fingerprinting), and identification risks from being introduced into the system. Because data is a valuable asset, some of the proposed solutions are not just directed at the user but also at the service (SP) provider, the cloud server (CS), or both, who may not want to divulge their data or certain predetermined locations of interest to the user.
Section 3.2, in particular, focuses on general privacy-preserving protocols that employ a single technique, such as homomorphic encryption, in preserving the privacy of the parties involved while delivering the LBS. In 3.3, we will look at the solutions proposed in the literature that are more relevant to this dissertation, particularly privacy-preserving protocols in an indoor context, which is

more challenging due to the complexity of computing the location coordinate. Finally, in Section 3.6, we look at some of the strategies that can be used to estimate the extent of privacy breach that occurs during the de-anonymization of information.

3.2 General Privacy-Preserving Protocols for LBS

In this section, we will examine a few instances of works published in the literature that make use of cryptographic algorithms to ensure the privacy of protocols, particularly those used in outdoor environments.

3.2.1 The Use of Homomorphic Encryption Techniques

We begin by examining some of the works that made use of homomorphic encryption, which is one of the most extensively used cryptographic techniques for achieving location privacy in location-based applications while still providing critical services to the user. The use of homomorphic encryption, an emerging technology, allows encrypted data to be securely processed by a service provider or a cloud server without the need to be first decrypted, allowing them to maintain the confidentiality of their data, which is particularly important in the wake of privacy concerns raised by regulations such as the General Data Protection Regulation (GDPR).
As a result, it enables a service provider (SP) and a cloud server (CS) to securely process and store the location data of users who query their services without compromising the location data of those users, thereby eliminating the need to choose between data security and usability when providing their services. This provides users with the assurance that they are not being tracked or that their data is not being used in any way other than what the protocol explicitly specifies. However, even if there is a data breach or a data compromise, the location data would still be securely protected because the data was never decrypted before being transmitted or processed. With the increasing demand for cloud services, the use of homomorphic encryption is becoming even more apparent, allowing mobile constraint devices to outsource the majority of their heavy computations to the cloud.

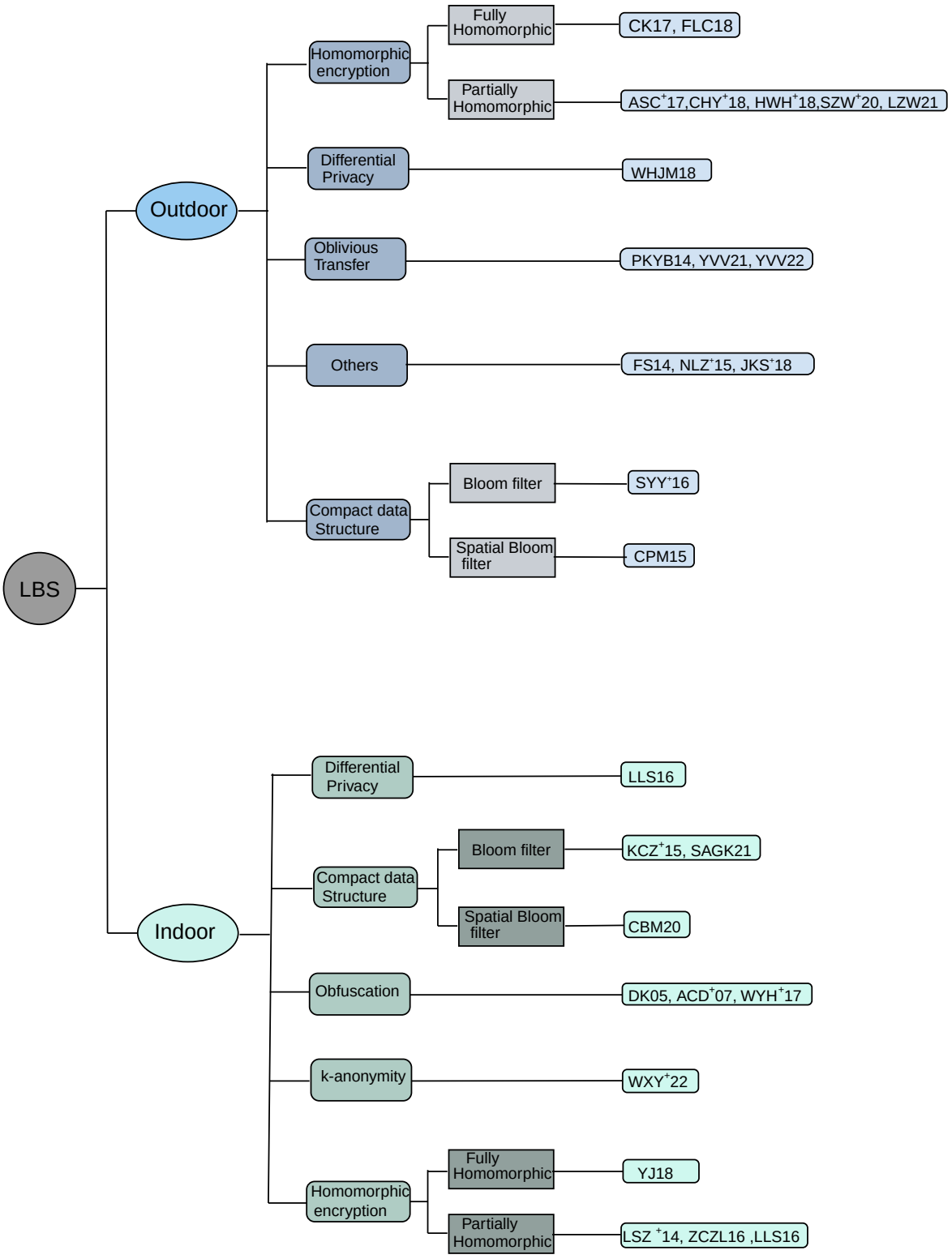


Figure 3.1: Taxonomy of Location Privacy-Preserving Protocols. The Techniques Under Consideration Are Classified According to Whether They Are Used Indoors or Outdoors.

There are different types of homomorphic schemes (partially homomorphic encryption, somewhat homomorphic encryption, and fully homomorphic encryption), and each comes with its own set of benefits and drawbacks when applied to LBS. For the purposes of this thesis, more focus will be on partially homomorphic encryption.

3.2.1.1 Partially Homomorphic Encryption

The work of [SZW+20] adopted Paillier encryption [Pai99] (Partially homomorphic) in preserving the location privacy of mobile users in what it's called fair meeting position determination scheme. For all parties, the scheme assumes an honest-but-curious (in that they follow the model's instructions) security assumption model. Their work asserts that both the service provider and the cloud server are responsible for computing the result of any request made by a mobile user, with the cloud server performing the vast majority of the computation on their behalf.
The protocol determines the fair meeting point by computing the geometric centre of the encrypted ciphertext sent by the mobile user. Thus, by utilizing the Paillier encryption, the CS is able to computes the geometric centre for n mobile users without knowing their location, ensuring the confidentiality of the location information.
In reference to [CHY+18], it presents an efficient privacy-preserving architecture that protects users' privacy while maintaining the efficiency of the protocol. Their work makes use of redundant point of interest (POI) data to secure user privacy against a location-based service provider (LBSP) or a semi-trusted third party. A proxy, that is, a semi-trusted third party (who adheres to the protocol execution's instructions), is then employed to filter out any redundant records (POI records). However, in order to avoid a breach of privacy on the part of the proxy and the LBSP, they develop a blind filtering mechanism that allows the proxy to filter out redundant POI without being aware of it. In particular, the system combines Paillier encryption with a cloaking technique (a lightweight randomization technique) to allow a semi-trusted party (proxy) to filter out homomorphically encrypted POI records without infringing on the user's personal information. Their security model makes use of semi-honest adversaries as well as non-colluding servers.
The Paillier cryptosystem is also used in the design of the k-anonymity technique for the location privacy preservation protocol by [LZW21]. The Paillier encryption is used in their work to securely filter out redundancy records oc-

casioned by the use of the k-anonymity technique. The scheme combines the Paillier encryption alongside the k-anonymity pseudo-random function (a deterministic function that generates a random output that is computationally indistinguishable from a random function within a polynomial time) in preserving the privacy of the protocol. Under the semi-honest scheme model, the proposed scheme claims to protect both the privacy of users' message queries and their location.

As stated in the paper, the work of [LZW21] represents a significant improvement over [CHY+18] in terms of computation and communication costs, and it is the most efficient solution for blind filtering, according to the authors. However, as discussed in [LZW21] unlike [CHY+18] the work of [LZW21] suffers from a complex LBS query (which includes the user's location, the query radius, and the query condition), though this comes with a caveat such as an efficiency penalty.

[HWH+18] presents a privacy-preserving location-based technique called "message in a concealed bottle "that protects the privacy of both parties (user and SP). Their scheme makes use of Paillier encryption to protect the location of the user as well as the SP's database while providing the LBS to the user. Specifically, it achieves so by employing Paillier encryption to determine the Euclidean distance between each attribute of a user's query and each attribute vector in the SP's database. Consequently, the user is able to participate in the computation of the search query without knowing the server's secret key. Because the Boolean circuit and Paillier encryption are used in conjunction, the user is able to acquire the desired point of interest (POI) while maintaining complete confidentiality about the query text.

Al-anwar et al. also present a privacy-preserving localization algorithm termed ProLoc in [ASC+17]. ProLoc is a privacy-preserving localization with private observers that uses partial homomorphic encryption. The proposed scheme allows observers (sensors or node anchors) to participate in the efficient computation of an accurate target's (location querier) location using the Paillier homomorphic encryption while maintaining the confidentiality of the observers' location and measurement. Specifically, the ProLoc algorithm allows observers with encrypted locations to participate in the estimation of a query node's location by transmitting their encrypted location to the aggregator without the observers disclosing their locations. Thus, the aggregator can estimate the encrypted location of the query node using just the ranges of encrypted locations received from the observers (sensors) without having prior knowledge of the encrypted

location of the query node (target's location) itself. As a result, the homomorphically Paillier's encryption cryptosystem allows only the querier (the query node) to decode the received encrypted location without knowing the locations or measurements of the observers involved in the computations. However, the proposed algorithm, ProLoc, is secure in the presence of honest-but-curious entities and when less than half of the observers are malicious (observers deliberately injecting false data). Additionally, the aggregator and the query node are non-colluding. The authors of [ASC+17] also claim that the proposed protocol's communication and computational costs are lower than those of similar schemes that use fully homomorphic encryption and garbled circuits.

3.2.1.2 Using Fully Homomorphic Encryption

Fully homomorphic encryption, which allows for arbitrary calculations to be performed on encrypted data, has emerged due to growing privacy concerns associated with outsourcing technology. On the other hand, the utilization of fully homomorphic encryption (FHE) comes with an extremely high cost. Moreover, when it comes to applications that require much computational power, the adoption of Fully Homomorphic Encryption is not commercially feasible, as demonstrated by Lauter et al. in [LNV11]. Nonetheless, we will look at a few of the technologies that have used fully homomorphic encryption to improve privacy.

The work [CK17] tries to solve the privacy concerns in location proximity by proposing a location privacy-preserving protocol for protection in location proximity. Specifically, their protocol uses the fully homomorphic encryption algorithm in single round trip computation without the involvement of any trusted third party. The proposed protocol, Locap (location information protection mechanism), allows a user (Alice) to request the location information of a nearby user (Bob) by sending her encrypted location information. This allows the responder (Bob) to homomorphically computes the radius distance without exposing the location information. Thus, using homomorphic encryption allows the two-party to compute their radius privately and securely.

Similar to [CK17], the authors in [FLC18] construct their private intersection scheme by employing the fully homomorphic encryption technique described in [Gen09]. The scheme allows one party (responder) to make a private and secure determination of whether the location information of the other party (querier) intersects any of the areas of interest of the responder. Their method uses a location server aggregator and applies the Bloom filter to obtain the level

of privacy that is required. In particular, the responder, Alice, encrypts her locations of interest and publishes them through the location server aggregator to prevent the querier (Bob) from accessing the plain location information. In a similar fashion, the querier encrypts his query using Alice's public key before sending it to the aggregation server. The location information of both Alice and Bob is then encoded in the Bloom filter. Subsequently, location matching is performed by the aggregator server to determine whether there is any location intersection between the querier and the responder. After decrypting the homomorphically computed location results, Alice will know if any intersection exists.

3.2.2 Solutions Based on Compact Data Structure Algorithms

Compact data structures are a way of storing data using as little space as possible for any given amount of data. They are designed to efficiently use the smallest amount of space possible in data representation while still allowing efficient query operations. Therefore, it is instrumental when considering low-end devices, as it reduces the resources needed for a large dataset in a distributed deployment.

The following work [SYY+16], presents a location method called BMobishare, which is a privacy-preserving LBS for a location-sharing mechanism that preserves privacy. Their work makes use of the space-efficient data structure, the Bloom filter, in masking sensitive data exchanges to prevent participating parties from obtaining unauthorized sensitive information. Their suggested solution entails four parties: users, a social network server, a location-based server, and a cellular tower. According to BMobishare's security threat model, all entities, with the exception of the cellular tower, are considered *honest-but-curious* in nature. In the scheme, the cellular tower is regarded as a trusted third party that has access to the location information of mobile users. The primary function of their location-based services system is to enable authorized users to query the nearby locations of their friends or strangers depending on their current position. This operates on the same model as other around-me services, such as locating the closest restaurant, drugstore, or gas station. The technique makes use of the Bloom filter in conjunction with two separate cryptographic encryption schemes (asymmetric and symmetric encryption), as well as an irreversible cryptographic hash function, to ensure that the privacy of the specified protocol is not compromised. In particular, the employment of a probabilistic data structure (the Bloom filter) ensures that sensitive information transmitted

between the two untrusted parties, the social network server and the location-based server, is kept private and secure.
In [CPM15], the authors present multi-party privacy-preserving protocols based on the byte array spatial Bloom filter, which is an extension of work done by the authors of [PCM14, CPM18]. The spatial Bloom filter itself is an extended work of the original Bloom filter that makes it possible for the client in the protocol of [CPM15] to privately computes their location while guaranteeing the privacy of the service provider's database. Thus, using the spatial Bloom filter helps preserve users' exact location in a locality.

3.2.3 Using Differential Privacy (DP) for LBS

There is an increasing interest in the application of differential privacy among researchers and large technological companies. In differential privacy, a controlled amount of noise is introduced into a dataset in order to prevent any sensitive information from being gleaned from the dataset or the original data from being extracted. This is done while ensuring that the dataset can still be utilised for its intended purpose, such as statistical analysis. Apple [1], for instance, provides users with Emoji suggestions or Lookup hints available by utilising differential privacy to protect the privacy of the participatory users. Google [2] is another example, one of the companies that pioneered the early commercial deployment of differential privacy and even released an open-source version of the differential privacy library.
Yufen et al.[WHJM18] present a differential privacy-preserving scheme for protecting users' location data. Their proposed algorithm, according to the authors, provides protection to the to-be-localized (TBL) client as well as the dataset of a location-based server. Their suggested protocol, DP3, employs a differential privacy technique in preserving privacy in a Wi-Fi fingerprinting indoor localisation system. Their scheme ensures the TBL's location is always protected. In particular, the scheme, which consists of four phases, masked the real location of the users by utilizing the differential privacy-enabled clustering algorithm by grouping the user-submitted access point (AP) vectors into k clusters and, subsequently, the location server permuting (based on permutation with exponential probability) the reference points in each of the k clusters. Because of the use of DP-based finger clustering in conjunction with Laplace noise, the approach prevents leakage of the server's location fingerprint.

[1] www.apple.com
[2] www.google.com

3.2.4 The Use of Oblivious Transfer Protocol in Location-Based Services

Even before the development of more powerful computing devices such as mobile phones or cloud servers, people have been using cryptography as a medium of secure communication for decades. Even before the advent of the internet, cryptography has been used to encrypt messages. In addition, an increase in mistrust among individuals who engage in business and daily contact has made the utilisation of cryptographic algorithms an even more useful practice. Furthermore, the use of oblivious transfer as a building block in multi-party computation makes it a very significant primitive in the field of cryptography. In their three different sets of LBS schemes proposed, authors of [YVV22] also made use of a two-party protocol known as the oblivious transfer (OT) to preserve the privacy of users of their protocol. Their first scheme is designed to protect user queries as well as the data privacy of semi-trustworthy cloud servers. The other two schemes protect both the user's query and the cloud server's location data, but the location cloud server is considered malicious in the security model.

[PKYB14] used the same oblivious transfer technique in their scheme for content protection of location queries. A two-stage approach similar to that proposed by [GKKB09] and [GKKB11] was used in their proposed scheme, with the first stage involving the use of OT as a solution, and the second stage involving the use of private information retrieval, as described in [PKYB14]. The algorithm protects the user's location by encrypting it with ElGamal encryption and employing oblivious transfer based on the Naor-Pinkas oblivious transfer [NP99].

The authors of [YVV21] offer a location privacy method dubbed linkable location-based services (L2BS) in an effort to address privacy concerns linked to the query content and points of interest (POI). Specifically, they combine oblivious transfer with a signature scheme to address the privacy concerns arising from query content and point of interest. Their protocol, L2BS, employs the OT algorithm to protect the Points of Interest of the service provider while preserving the privacy of the users' query information. According to the designers of L2BS, their technique enables the SP to seamlessly connect previously rendered services to returning clients while maintaining their privacy and incurring a minimal computational expense.

3.2.5 Additional Proposed Solutions for Location-Based Privacy-Preserving Protocols

The following set of proposed solutions takes into account some of the approaches not yet covered in this thesis. These techniques have also proven to be useful in modern applications on a wider scale. Specifically for Google Android users, the authors of [FS14] presented a location privacy protection technique that was tailored and designed exclusively for them. Their protocol, dubbed LP-Guardian, which was implemented and evaluated on Android users, is said to address some of the shortcomings of existing approaches by addressing some of the most prevalent threats related to location tracking, user profiling, and user identification while maintaining the functionality of the proposed application. According to the authors, the proposed scheme, LP-Guardian, foils location privacy risks without impairing the end-user experience of the application at an overhead cost of less than 10%. Additionally, the end-user has greater control over her important location because she selects which places are more relevant and which ones are less essential to be used for user profiling purposes, if any. The threats model of this LP-Guardian application is based on the honest-but-curios adversary. Though, albeit relaxed threats model, the authors argue that the adversary may just be interested in only inferring the end-user location information through the available location APIs. In all, the evaluation of the LP-Guardian was performed on three-different datasets containing 100 users from three different cities collected over a period ranging from a week to a year.
Several publications [ZCN+13, STP+14, NLZ+14, NLZ+15, ZCLW18] consider the necessity of caching as a technique for ensuring the protection of privacy of location-based services. Because query data can be cached and resubmitted for similar requests, it can occasionally improve privacy, particularly when interacting with untrustworthy LBS servers. A caching approach similar to those provided by [ZCN+13, STP+14] was offered in [NLZ+15] in order to protect the privacy of users in location-based services and also assessed to what extent caching can improve location privacy. Their approach maximizes the location privacy of a current querier by adopting an entropy-based strategy that incorporates caching for privacy protection. Furthermore, they presented two caching-aware dummy algorithms: CaDSA and enhanced-CaDSA. The former improves location privacy by picking queries with equal probabilities to achieve k-anonymity, while the more thorough latter takes into account data freshness and caching speed to improve location privacy. According to their findings, this proposed method improves privacy when compared to earlier caching-aware

and caching-oblivious strategies.
The method proposed by [JKS+18] aims to handle privacy pertaining to proximity detection, such as those related to nearby pharmacies or friends who are located nearby. To accomplish the intended goals, such as bandwidth and faster runtime, their suggested technique, privacy-preserving location proximity (PPLP), incorporates numerous secure computation protocols. In particular, the proposed method [JKS+18], which is based on the Euclidean distances, makes use of a number of different cryptographic protocols. These include secure multiparty computation, the DGK encryption, elliptic curves, ElGamal encryption, and Bloom filters. Their scheme, which was evaluated on a real-world mobile setting, is said to have a total runtime of below 200 ms and online runtimes of below 50 ms.

3.3 Privacy-Preserving Protocols for Indoor Environment

In the following part of this section, we will discuss new research that has been conducted on the topic of protecting individuals' privacy when it comes to location data in an indoor setting. This can be further broken down into homomorphic encryption, secure multi-party computations, compact data structures, and server-aided protocols.

3.3.1 Homomorphic Encryption (HE)

Li et al. [LSZ+14] introduced a protocol for privacy-preserving indoor localization that makes use of homomorphic encryption to ensure that the privacy of both parties is maintained during the process. Their approach, known as the privacy-preserving Wifi Fingerprinting localization scheme (PriWFL), in particular, employs additively Paillier encryption to secure end-user location privacy as well as the service provider Wi-Fi fingerprinting database. In their work, they presented two types of protocols: a naive version of the protocol and a complete version of the protocol. In terms of security, the complete version offers a significant improvement over the naive version in terms of the security guarantee of sensitive information for both parties. However, the security model for both protocols only takes into account a semi-honest adversary in protecting both parties' sensitive information.
The work of [YJ18] presented a privacy-preserving protocol for indoor Wi-Fi lo-

calization, which is an improvement over the security guarantee of the protocol discussed in [LSZ+14]. Given the vulnerabilities [YJ18] identified in the protocol of [LSZ+14], an adversary will be able to carefully craft a packet (query) that will compromise the service provider's Wi-Fi fingerprinting database. As discussed in [YJ18], the attack is based on the fact that all calculated distances are masked using the same randomness $(\mathcal{R})$, making it possible for an adversary to remove the randomness by simply subtracting two masked distances. In an effort to mitigate the risks, a number of solutions were proposed in the paper, including homomorphic encryption, garbled circuit, and a combination of the two algorithms.

Using homomorphic encryption (again based on the Paillier cryptosystem), Zhang et al. [ZCZL16] take a more advanced approach to protect the untrusted parties involved in the protocol. They employ a support vector machine (SVR), a machine learning algorithm for classification and regression, to protect the untrusted parties involved in the protocol. Based on the Wi-Fi-fingerprinting positioning technique, the authors developed a privacy-preserving system for indoor localisation. Their solution makes use of the Paillier cryptosystem to ensure the location privacy of the end-user as well as the privacy guarantee of the server's database, are preserved. Even this protocol, however, is not secure against a malicious client: a specially constructed query by a client can readily reveal the decision function used by the service provider in training the data set.

With the use of the Wi-Fi fingerprinting approach, the researchers in [LLS16] present a privacy-preserving site survey scheme that is designed to be secure as possible and private. In order to protect the participants in the site survey scheme, the work makes use of both homomorphic encryption and differential privacy techniques. In particular, the proposed scheme employs Paillier homomorphic encryption to protect the location privacy of suppliers (users recruited by service providers to send Wi-Fi signals) who participate in the crowdsourcing Wi-Fi fingerprinting-based survey while also ensuring the usability of the aggregated data. As stated by the authors, all of the suppliers participating in the scheme work together to conceal their measured data from the service provider by utilizing the Paillier homomorphic encryption algorithm. Using differential privacy on the released dataset (measured data), the authors ensure that the location privacy of the supplier is not compromised regardless of whether a supplier is present or not in the group of the outsourcing site survey, as described in the paper.

Next, we will explore a significant data structure algorithm extensively used in privacy preservation, notable for its key features and low computational complexity.

3.3.2 Privacy-Preservation Based on Compact Data Structure Algorithms

When it comes to location-based services, the usage of compact data structures has been recommended as a means of reducing the risk of end-user privacy being compromised. The Spatial Bloom Filter (SBF) and the Counting Bloom Filter (CBF) were described by the authors of [CBM20] as two compact data structures to be utilized in conjunction with one another as location information processing technique when used in an indoor context. With their privacy preservation technique, the authors aim to reduce the gathering of personal user location data, which leads to the prediction of user behaviour. For the evaluation of their proposed algorithm, the authors relied on the Cisco Meraki network infrastructure to gather their location dataset in an indoor environment. According to the authors, the data structures they used (the Spatial Bloom Filter (SBF) and the Counting Bloom Filter (CBF) performed admirably in terms of memory consumption and computational cost. However, their work security model relied on the trust of a data anonymizer who handles the communication between the Data provider (DP) and an honest-but curious Data consumer (DC) who is in charge of analysing the streams of data acquired by the data provider to infer personal information.
In [SAGK21], the authors propose a Wi-Fi fingerprinting privacy-preserving localization technique that makes use of a Bloom filter in a setting where an untrusted and curious LSP party is present. The proposed scheme enables the user to compute its location anonymously by utilizing a partial radio map (PRM) and with the assistance of a location service provider (LSP).
To be more explicit, the user selects the first access point (AP) with the highest RSS value and then applies a Bloom filter to mask the media access control (MAC) address of the selected AP. In order to protect the user's identity, the user submits the Bloom filter vector she has generated to an anonymizer (a non-colluding entity that functions separately and independently of the LSP), which then forwards the user's Bloom filter vector to the LSP for processing. As a result, the LSP is able to compute the user's PRM anonymously. When compared to earlier works that have employed comparable methodologies, the

authors assert that their technique has a lower level of computational complexity. According to the authors, this can be attributable to the usage of Rivest-Shamir-Adleman (RSA) encryption and double encryption in those previous works. Their work's security model, on the other hand, solely took into account a trusted anonymizer and a semi-trusted Location Service Provider, both of which are independent and non-collaborating servers.
The Bloom filter is also used by the authors of [KCZ+15] in their privacy-preserving indoor location protocol for smartphones. The authors' suggested approach, dubbed Temporal Vector Map (TVM) scheme allows a mobile user (smartphone) to precisely localize her location by using a k-anonymity Bloom filter. Consequently, the user can localize through an Indoor Positioning System (IPS) entity (i.e., server) that is considered to be untrustworthy, while at the same time, the user is able to maintain location privacy thanks to the TVM. In detail, a user participating in the scheme computes a k-anonymity Bloom filter for each request to an Indoor Positioning System using a cryptographic hashing algorithm and user-defined preference of, say $k = 3$. This, for example, will make it more difficult for the IPS to differentiate the user's location request from those of *k-1* other users. As a result, if a user makes a location request from building 1, the server will presume that there are two more such requests from buildings 2 and 3. The user achieves this level of *k-anonymity* protection by employing the nearest neighbours technique to generate a series of disguised localization requests that are identical to the user's initial location request. With this technique, the server is prevented from determining the user's location after k consecutive independent continuous localization requests. One of the primary drawbacks of this scheme is that a portion of the IPS's database is transferred to the user for localization purposes, making it expensive for resource-constrained mobile devices.

3.4 The use of Obfuscation Techniques in LBS

Obfuscation techniques used in location-based services are basically methods or ways to obscure or hide the precise location of a user or device (sometimes in the form of a user's query). These techniques can be employed in a number of different ways. The primary purpose of these techniques is to protect the users' privacy, which ultimately results in the techniques' conformance with the applicable privacy regulations. The techniques typically reduce the accuracy of the location information; for example, adding random noise, using geofencing,

or decreasing the accuracy of the radius rather than the exact location point. Other ways include using k-anonymity or randomization to modify the location data. Though location obfuscation comes with a number of advantages, including increased privacy, user trust, and regulatory compliance; however, it also has some disadvantages, including a reduction in accuracy and functionality. For instance, location services that make use of geofencing strategies to limit the access of location data may be unable to provide certain location services beyond the geofence.

According to [DK05], location privacy protection using obfuscation-based approaches plays a crucial role in protecting users' location privacy, particularly in a pervasive computing setting. Their suggested framework strikes a balance between user location privacy and improved quality location information services by purposely degrading the quality of the user's location information in the context of enhanced quality location information services. This is intended to allow location-based service providers to obtain just the relevant location information from participative users in order to deliver a tailored service to those participatory users. As a result, the suggested solution addresses the issue of maintaining the privacy of users while yet providing the necessary location-based services. It is important to note that the downside of this protection mechanism is that the utility of quality location services is always compromised in exchange for location privacy, a trade-off between utility and privacy.

To mitigate privacy leakage in mobile crowdsensing applications, particularly for individuals who are never get selected for any assignment, [WYH+17] presents a location-based crowdsensing task model. The model, which is based on the technology of geo-obfuscation, preserves users' locations during job allocation. Using differential privacy, the scheme allows users to conceal their reported locations from their adversaries without the involvement of any trusted third party. As a result, users' reported locations are protected regardless of whether their adversaries have prior knowledge of them. As the primary objective of mobile crowdsensing (MCS) is to maximise the user experience while also guaranteeing that all tasks inside the sensing area are fulfilled. In light of this, the authors of [WYH+17] suggested a solution in what they call mixed-integer non-linear programming to reduce the travel distance of the selected users based on the Benders Decomposition, which is well-known for its ability to handle large scale linear programming problems. The proposed method, which surpasses state-of-the-art Laplace obfuscation, was evaluated on both simulated and real-world mobility traces, according to the authors.

[ACD+07] proposes a privacy-enhanced spatial obfuscation-based framework with a varying degree of location privacy that also takes into account the accuracy of location information. Their framework first allows users to indicate their privacy preferences in a simple and coherent way based on their location information. Then based on this user privacy preference, they implement one of their three proposed obfuscation techniques to preserve user privacy.

3.5 Server-Aided Location Privacy-Preserving Protocols

The usage of resource-constraint devices (IoT) has become increasingly helpful, especially in the sharing of information among these devices, as a result of the expanding trend and increased demand for automation. However, it comes with a variety of drawbacks, including those related to storage, power, communication, and processing capacity, as well as scalability. With these constraints in mind, it has been highly sought after to address these obstacles. The utilization of resource-rich cloud computing has become increasingly popular as a means of decreasing the load on these resource-constrained devices while simultaneously resolving any privacy concerns that may exist. Ongoing efforts to address some of the growing privacy concerns linked with the ever-expanding convergence of resource-constrained devices and cloud servers are being pursued by a number of research projects.

According to the research of [WXY+22], their privacy-preserving indoor localization system made use of an outsourced cloud service. Their indoor localization technique is based on Inner Product encryption, which is supported by a cloud server that provides assistance. It makes use of the cloud server to estimate the private location of users without obtaining access to the users' actual positions, thereby protecting the location privacy of users who participate. For the server to be able to make a secure estimation of the user's location, the server makes use of the computability of the Inner product encryption on the ciphertext. In particular, the anchor information (user location information) is encrypted during the setup phase (offline), and then the distance difference between participating users is encrypted and transferred to the assisted cloud server during the online phase. In order to conceal users' true location, their proposed scheme makes use of the k-anonymity technique, by encrypting the difference in distance information between users, in addition to using addi-

tional decoy vectors (information). The authors claim that by incorporating three optimization strategies into the location estimation algorithm at the cloud server side, they have been able to increase the performance of their protocol greatly. However, in estimating the location of users, their protocol only takes into account honest-but-curious cloud servers. Additionally, the communication and computation complexity associated with performing Inner Product-based encryption computation is very expensive in its current state.

3.6 An Overview of De-anonymization Techniques in Location Datasets

In this section, a comprehensive overview of the many different de-anonymization attacks, particularly those that target location datasets, is presented. In particular, we show the various techniques of de-anonymization that the adversary investigates to acquire meaningful information (background knowledge) about the users who are the targets of the attack, as well as the attack surfaces that are investigated to achieve a successful de-anonymization. But, first, we will describe a high-level framework for de-anonymization attacks that covers a variety of methodologies and kinds of attacks used.

3.6.1 Location Dataset

Location data is any data that represents the geographical positions of mobile devices such as smartphones, tablets, smartwatches, Internet of Things devices, and so on. The geographical positions, also known as coordinates, are usually expressed in latitude and longitude. Aside from the coordinates, which define the granularity of the location data, other features include altitudes/elevation (height above sea level), timestamps, IP addresses, device IDs, and many more. There are several used cases of location data, including proximity-based services such as weather and navigation, location-based advertising and analytics, traffic and local search, and even social networking. In addition, location data makes it possible to track the movements of things or people to understand their pattern behaviour better. Furthermore, the use of location data can help improve the effectiveness of customer campaigns. When it is possible to link the location data directly or indirectly to a living individual, it is regarded as personal location data according to the definition provided by GDPR[88]. All of these uses of location data present grave privacy concerns, as the granu-

larity of the acquired data, can reveal the religious views and health status of individuals.

3.6.1.1 Sources of Location Data/Collection of Location Data

Many individuals are of the opinion that location data is essentially the same thing as Global Positioning System(GPS) data. However, the reality is that GPS data is just one of many sources that may be used to acquire location information. In addition to the Global Navigation Satellite System (GNSS), of which GPS is a component, there are a number of other systems, including cell towers (which make use of a technology known as triangulation to estimate the location of mobile devices), Wi-Fi Networks, RFID, Bluetooth beacons, and plenty of others. Each of these sources comes with its own set of benefits and drawbacks; for instance, GPS works well when utilised outside, whereas Wi-Fi tends to have stronger signals inside.

3.6.2 Location De-anonymization

In this section, we will examine a variety of de-anonymization attacks that have been reported in the literature under review. The process of deanonymization is essentially a form of reverse engineering that makes it possible for an adversary to successfully re-identify an anonymous dataset or dataset previously thought to be anonymous. In their renowned publication on the Netflix prize dataset, Narayanan and Shmatikov [NS08] contributed extensively to this field.
We will thus consider a variety of approaches, as well as the attack vectors that are utilized during successful attacks. Therefore, when we refer to an approach, we are referring to both the methodology that the adversary utilized or relied upon in order to build the background knowledge of the target dataset as well as the de-anonymization attacks that were employed in accomplishing successful de-anonymization attacks. To begin, we provide an illustration of a high-level overview of the various attack methodologies alongside the type of target dataset that was employed based on the various literature papers that were sampled.

3.6.2.1 Target Dataset

The target dataset normally refers to the source of the target dataset (anonymized), which the adversary is interested in de-anonymizing. At this

stage, the adversary deems this target dataset as containing sensitive information of interest. The target dataset may be freely accessible or restricted from public access. Many times, when the owners or data collectors of these datasets classify the dataset as anonymized, they are typically referring to the removal of personally identifiable information (PII) such as names, Social Security numbers (SSN), cell phone numbers, home addresses, email addresses, and other similar identifiers. In order for an adversary to be successful in de-anonymization, special attention must be paid to the data structure, the anonymization technique used on the dataset, and, in particular, the de-anonymization technique to use. This is because each of these factors has the potential to have an impact on how successful the de-anonymization is [AMS+15]. For example, the GeoLife dataset and the Crowsad dataset are two completely different independent datasets with different resolutions, even though they are both location datasets. That is, a technique that may work for one dataset may not work for the other with the same parameters.

3.6.2.2 Auxiliary Data/Information

On the other hand, auxiliary datasets or side information are representations of the background information that an adversary constructs in order to aid in the de-anonymization attack on the target database. This auxiliary, or side information, is not anonymised in most cases. However, this is not always the case and can vary depending on how the auxiliary information is collected. Auxiliary information can be categorised as either internal or external. The formal, also known as intra-dataset, refers to the dataset extracted from the same target dataset. This can be subdivided into two subcategories: subset auxiliary dataset (the extracted side information still exists in the original target dataset) and disjoint auxiliary dataset (the extracted side information is no longer contained in the target dataset, i.e., the auxiliary and the target dataset are disjoint). The latter, however, is side information that comes from an entirely separate source rather than the target dataset.

3.6.3 De-anonymization Techniques

One of the most important aspects for the adversary to fulfil its de-anonymization objectives is the auxiliary information that the adversary uses to map the target user to the target database. Before mapping/re-identification can be performed, the adversary must have a high-level overview of the target

trajectory record to construct the necessary profiles for the target users. Only then can the mapping or re-identification be done. Simply having access to the target dataset is insufficient to de-anonymize a target user in the dataset, even if the target user's auxiliary information is also available. As a result, a list of distinct profiles needs to be derived from the target dataset to facilitate mapping the generated auxiliary (side) information to the constructed profiles derived from the target dataset. As mentioned earlier, sometimes the side information might be derived from the same target dataset and can also be in the same data structure as the target dataset. In many cases, not only is it necessary to determine the strategy that will be used for the attack, but the end goal is also essential and is normally known in advance. Therefore, in order to successfully de-anonymize a target user, it is essential to have a solid understanding of the goals of the attacks being launched against the target users. Linkage and inference attacks are two examples of well-known types of attacks found in the body of academic research. Consequently, numerous strategies for de-anonymization lead to either of these types of attacks (linkage or inference) or sometimes both of them. In the following, we will investigate some of these adopted approaches in the literature.

3.6.3.1 Mobility Models

In their most basic form, mobility models are mathematical constructs that simulate the behaviour of people's movement patterns. When applied to individuals' mobility trajectories, these mobility models provide the adversary with tremendous information regarding the individuals in the target database. In theory, the adversary may not know the identity of individuals whose location information is contained in the target dataset. However, the use of mobility models allows the adversary to have a solid understanding of the mobility patterns of the individuals, including sometimes sensitive information such as their place of worship or their health status. As a result, it allows the adversary to know and understand the dataset's characteristics very well, though the identity of the individuals within the target dataset is unknown at this point of the attack.

The mobility model has been used in a number of publications, including [MDBP08, STBH11, GKdPC14, WZF+15, MKH17, EP22], to carry out de-anonymisation attacks successfully. For instance, Mulder et al. [MDBP08] use a mobility model to re-identify mobile users within a cellular network. Their linkage attack employed the Markovian mobility model to build pre-existing

profiles of the target users using their movement on the GSM network. These built profiles were then used to perform the attack. Similarly, Gambs et al. [GKdPC14] use the mobility model, MMC, the Mobility Markov model, to de-anonymise users in an anonymous mobility dataset. Their Markov model depicts the states by taking into account the locations in the target dataset that users most frequently visit. Their attack is categorised as part of the linkage attack in the survey conducted by [FKZ+20], despite the fact that the original paper [GKdPC14] refers to it as an inference attack. Based on the findings of the original paper [GKdPC14], it can be concluded that the attack model, which was developed to utilise important places for state transitions, is effective for both inference and linkage. In a similar manner, the research conducted by Shokri et al. [STBH11] utilized a mobility model to analyze location privacy protection mechanisms (LPPMs) in order to quantify location privacy. In particular, they developed a generic model that can also take the shape of a Markovian chain model. This was done to simulate the adversary's background knowledge when performing the de-anonymization attack. Consequently, the adversary transforms the target users' movement traces into events, which are then encoded. For example, repeated visits to the home in the evening could increase the weight of such an event. Their evaluation was based on a sample of 20 mobile users (vehicles) chosen randomly from the CRAWDAD [PSDG09] mobility dataset collected over the San Francisco mobility bay area.
The identification of users in the GSM network was reconstructed by Mulder et al. [MDBP08] by using a Markovian model in which static cells represented the states of the Markovian model. As a result, users tend to register their new locations whenever they enter within a certain radius of a new cell. With this updated information, mobility profiles of the target users are developed to depict their mobility trajectories to serve as the basis for the de-anonymization process.

3.6.3.2 Areas of Interest (AoI)

Another technique mostly adopted in de-anonymization attacks is something that we refer to as areas of interest or important locations. These are locations that a user or target individual usually visits and spends considerable time on. Consequently, the mobility traces of an individual will have a significant number of these places, given the activities that such an individual engages in on a daily basis. Examples of such places includes place of worship or work. Numerous studies [Kru07, FSH11, ZB11, GLP+13, RM14, NUTV16] have utilized this

technique in building the auxiliary knowledge for the de-anonymization attack. This strategy for re-identifying users was effectively employed by Freudiger, Shokri, and Hubaux [FSH11] in the course of their geolocation attack. In fact, their work considered both spatial and temporal influences in their attack. The authors adopted a pair of home/work as their pseudo-identifiers to perform the attack. The locations that users frequently visit between the hours of 9 am and 5 pm are referred to as work, while the locations that users usually visit overnight between 9 pm and 9 am are regarded as their homes if they spend a significant period, say, an hour. They examined the efficiency of this method by applying it to two distinct mobility datasets in two cities, namely Borlange and Lausanne. The mobility traces from the city of Borlange only included 24 users and were gathered over a period of two years (1999-2001), whereas the mobility traces from the city of Lausanne contained 150 users, which were captured over the course of one year. Their findings offer some evidence to support the hypothesis that the Spatiotemporal correlations of mobility traces are frequently unique to individual users within a target dataset.

Zang and Bolt [ZB11], on the other hand, defined important locations as the top n most frequented locations in their research (Anonymisation of Location Data Does Not Work). In particular, the authors qualified the top n most visited places in a GSM network by an individual as a quasi-identifier of the anonymisation attack. Their study, which was carried out throughout all fifty states in the United States, was based on approximately 30 billion Call Data Records (CDRs) with about 25 million cell phones (and roughly 20 million unique individual users for the study). These Call records were collected over three consecutive months in the year 2010. Furthermore, their findings indicate that, based on the top two most frequented locations, 35 percent of the users (out of 20 million subscribers) might be re-identified within two-week of mobility traces. In addition, when traces spanning more than two weeks were analysed, over half of the population under study could be re-identified based on the top two areas they frequented the most (GSM cells). The authors concluded that location data collected over a prolonged period potentially threatened individuals' right to privacy.

Similar to [ZB11], the research presented in [NUTV16] takes advantage of the behavioural pattern of users in a GSM network to carry out de-anonymisation attacks. These attacks are carried out by analysing the statistics of the users' mobility behavioural patterns. Specifically, in performing the de-anonymisation attack, the authors categorised the users' locations as histograms (the total

number of visits within the specified regions and time frame) of unknown and known users. These locations, also known as important locations, are, in essence, the area range within a GSM cell tower that was captured during the data collection period. The resulting histogram for each user then serves as the user's fingerprint for carrying out the de-anonymisation attack. The users who were formerly labelled as unknown (anonymised) and known users are then matched using the histograms produced for each set. Their study demonstrated the effectiveness of this method by analysing Call Data Records (CDRs) of anonymised phone records from around 50,000 people in the Ivory Coast over two weeks. According to their findings, anonymised users can be re-identified with high accuracy using only their statistical behaviour.

3.6.3.3 Random Locations/Spatio-Temporal Points

Now, we take a look at another important feature (random points or locations) that has been widely employed in the literature [MYYR10, DMHVB13a, DMRSP15, WGL+18][4,7,9] to build adversary knowledge with respect to the auxiliary information. These Spatiotemporal points reflect a pattern of location coordinates with associated timestamps, making them distinct within the target dataset. This pattern of uniqueness is what the adversary normally exploits to perform the de-anonymization attack. Thus, a small number of these distinctive Spatio-temporal points are all that is needed to pinpoint a target user in a dataset, even in the absence of personally identifiable information (PII) such as a name, social security number, or other similar details.
This method was utilized in the research conducted by De Montjoye et al. [DMHVB13a], which aimed to quantify the uniqueness of an individual's mobility traces in a crowd of approximately 1.5 million people. For the purpose of the study, a dataset consisting of mobile phone calls and text messages was employed. The study recorded the location of the GSM network antenna that each user connected to when making a call or sending a text message. The period covered by the dataset collection was from April 2006 to June 2007. Their research demonstrates that an adversary's knowledge of users' mobility patterns at two random locations (two Spatio-temporal points) is sufficient to uniquely identify fifty percent of the anonymized users in the dataset. The result is exacerbated when four Spatio-temporal points are considered, allowing the adversary to uniquely identify ninety-five percent of the anonymized users in the dataset. Their work shows that an adversary's knowledge of users' mobility patterns is just enough to successfully re-identify the users in the target

database. Even though the dataset utilized for the research was not a disjoint set (there was no distinction between the trained dataset and the tested dataset), the authors concluded that even when a coarse dataset is used, very little privacy could be maintained.

The utilisation of metadata provides additional support for the conclusion presented in [DMHVB13a] of the work that De Montjoye et al. [DMRSP15] released. Their research [DMRSP15], analysed credit card transactions for approximately 1.1 million users for three months. The authors concluded that using four Spatio-temporal points (shops visited by the users per transaction per day) is just enough to re-identify 90 % of the users involved in credit card transactions. It is interesting to note that the authors claim that the chance of re-identification is increased by approximately 22 % when the approximate price of the product that the customers bought is included in the Spatio-temporal points. In addition, they concluded that women had a greater propensity to be re-identified than men because of the greater number of stores that women frequent compared to men. In a similar vein, individuals with higher incomes have a greater re-identification risk in the context of credit card transactions, which poses a more significant threat to their privacy.

The work of [WGL+18] also used randomization of Spatio-temporal points (locations) to perform de-anonymization attacks. In addition, their work made use of a large-scale ground-truth mobility dataset for their study. One trajectory data collected involved about 2,161,500 users of GSM cellular networks, with the other two being external datasets of social networks of over 57K users of GPS check-in traces and the other being GPS check-in services of about 46K users in total. Based on the service provider's provided database, the authors constructed the mobility traces for users by converting GSM cell towers (base stations) to their specific location coordinates (longitudes and latitudes). From their analysis, the authors concluded that to guarantee location privacy for users in a location dataset, privacy protection mechanisms should not factor only into the user location patterns but also the location context.

Chapter 4

Efficient De-anonymization Attacks on Location Data Using Hidden Markov Model

In this chapter, we design and implement de-anonymization attacks for geolocation datasets, utilizing the hidden Markov model and user mobility profiles. The findings described in this chapter were presented in Genoa, Italy, and included in the Proceedings 2022 IEEE European Symposium on Security and Privacy Workshops (EuroSPW).

4.1 Introduction

The widespread adoption of smart devices equipped with geolocation sensors has led to the increased availability and use of location-based services (LBS). Such services enable service providers (SP) and telecommunication companies to collect large mobility datasets, often spanning millions of users who subscribe to these services. These mobility datasets are particularly valuable, as they can be used for urban planning, traffic forecasting, network optimization, targeted advertising, and a variety of other purposes, including contact tracing in the recent Covid-19 pandemic [CMP12, LXB+21]. Service providers have, therefore an incentive to publish or sell such datasets. A 2020 Grand View Research report on the market for personal location dataset estimates the market value at USD 10.6 billion in 2019 and predicts an annual growth rate of 15.2% over the forecast period (2020-27) [GVR20]. The growth of open data policies across the world has also led to an increase in the freely available mobility datasets [FKZ+20].

Due to privacy concerns regarding sensitive information that can be extracted from individuals' mobility data in published datasets, owners of mobility datasets are often required to replace personal identifiers such as name, address, and social security numbers (SSN) with pseudorandom identifiers to make the dataset (pseudo) anonymous. However, privacy risks still exist even if personal identifiers are removed before the mobility dataset is published or sold, and a large body of research has demonstrated that the personal data of individuals are still at risk of exposure, even if their personal identifiers are removed [FSH11]. For example, the analysis of places and geographical areas that are frequently visited by pseudonymous individuals in a dataset (areas of interest) can reveal their home, workplace, and place of worship [FSH11]. Moreover, side information such as social networks geo-tagged posts (on platforms such as Twitter and Instagram), and publicly available mobility trajectories databases can easily be used to de-anonymize individuals whose data is included in the mobility dataset [SH12]. This is true even in the case of very large scale datasets: the work of Kondor et al. [KHdMR20] presented a large-scale re-identification attack using mobile networks and transportation card usage, achieving a success rate of about 55% over four weeks of activities. Recent work by Farzanehfar et al. [FHdM21] confirms that the threat of re-identification persists even in a country-scale mobility dataset.
Even without secondary information, depending on how the personal identifiers are removed or how the dataset is sanitized, retrieving the auxiliary information from the same target dataset is sometimes feasible. This was illustrated in the work of De Montjoye et al. [DMHVB13a], who mined anonymized users' mobility records to uncover mobility patterns associated with the individuals in the target dataset. Another technique uses the home/work locations couple as a quasi-identifier to re-identify the users in the anonymized mobility dataset [FSH11].
Given the various re-identification attacks proposed in the relevant literature, it is difficult to estimate the actual level of privacy (if any) achieved by anonymizing a dataset. For this purpose, Shokri et al. [STBH11] developed a framework for quantifying location privacy. Their works address the difficulties inherent in comparing various location privacy protection mechanisms (LPPMs) due to the lack of a systematic process. The work of Wang et al. [WGL+18] makes a case for a performance mismatch between algorithms and theoretical privacy bound. They ascribed the discrepancy to an underestimation of the impact of spatial-temporal data from a variety of sources. The performance of these algorithms

was evaluated using two real-world datasets and then proposed algorithms to improve the performance.

Most de-anonymization attacks on geo-located databases, including those discussed previously, usually consider the user's temporal or spatial mobility trajectories (or both) in de-anonymizing or inferring sensitive information from the target database. These attacks are generally categorized as either linkage attacks or inference attacks. In the former, separate user accounts that belong to the same individual but are either anonymous or under different usernames are *re-linked* through data analysis. In the latter, an adversary tries to reveal personal identifiers or sensitive information of anonymized records in a target database using secondary information (auxiliary information) different from the target database or hidden values from the same target database.

In general, to carry out a de-anonymization attack, one must first decide on the most appropriate strategy for gathering background information on the target individuals (user fingerprinting) in the target database. The adversary then decides on the type of de-anonymization technique (for example, feature matching, graph-based, mobility models matching, statistical matching, etc.) that best suits the background knowledge modelled after the user fingerprinting. Finally, the de-anonymization attack is performed using the chosen de-anonymization technique. The success rates of these techniques are normally quantifiable and, as discussed in [AMS+15], the combination of the type of attack technique chosen and the background knowledge has an impact on the success of the de-anonymization attack.

Among the various de-anonymization techniques that have been presented in the research literature, a number of them are based on the hidden Markov model (HMM), a statistical approach that is frequently used in linear sequence 'labelling' problems [DCC09]. In the context of location, the HMM approach has proved effective in extracting hidden information from complex trajectories [WGL+18, WZF+15]. Generally, modelling HMM to build background knowledge on the target database mainly involves two phases. First is the use of a clustering algorithm to identify the areas of interest to feed into the model. For instance, if the areas of interest represent the hidden states, identifying them from the raw mobility dataset usually requires a clustering algorithm (such as K-Means, DBSCAN, Hierarchical, etc.). In line with this, the Density-based spatial clustering of applications with noise (DBSCAN) algorithm is used in this work. The second phase is to compute the transition probabilities based on the hidden states transitions, and the emission probabilities using the observation

sequence.
A successful de-anonymization threat usually leads to identity disclosure, linking two or more de-anonymized accounts of different networks to the same user and disclosure of sensitive information such as an address, email, SSN etc. [SGB12] identifies and classifies the above three primary threats emanating from a successful de-anonymization attack, and names them *content* (disclosure of sensitive data inferred from the location), *linkage* and *identity* (full re-identification of the user).

4.1.1 Contribution

In this work, we propose two novel Hidden Markov Model-based attacks that take into account both the temporal and spatial influences on user mobility trajectories. Our model, in particular, provides initial predictions about the user's regions of interest and, based on these predictions, constructs a hidden Markov model that outputs the probabilities of a user visiting any of these areas of interest for some observed given days of the week. Together, this process sums up the user's profile (which we describe as user fingerprinting) to re-identify the target individuals. Based on this user profiling, we propose and implement two de-anonymization attack mechanisms: the *Forward* algorithm and a divergence measure algorithm (based on the Kullback-Leibler divergence).
Finally, we evaluate the performance of the de-anonymization algorithms using real-world mobility traces from two different publicly available datasets (GeoLife [ZZXM09] & Taxicabs [BBL$^+$]). Experimental results show that our proposed de-anonymization algorithms achieve a re-identification success rate (over 80%) that is significantly higher than those achieved by comparable studies, over the same dataset.

4.2 Preliminaries

This section presents preliminary information that is useful in the understanding of the remainder of the chapter. In particular, we present the main techniques that are used as a basis for the construction of the novel de-anonymizers proposed in this work.
Two main techniques are used in this chapter for building the mobility behaviour model of users: DBSCAN (presented in Section 4.2.1) and HMM (Section 4.2.2). In particular, the DBSCAN, a clustering algorithm, is used to cluster

Table 4.1: Notation: Acronyms and Symbols

Symbol	Meaning
N	the total number of states
n	the total number of users
π	initial probability distribution over all states
$\triangle$	area of Interest or states
δ	the days observation sequence of size T
$\beta_{i,j}$	the emission probabilities of moving to state $\triangle_i$ of type δ_j
γ	the transition probability matrix with members represented as γ_{ij}
T	maximum size of observation sequence

each individual's areas of interest. In other words, it is an unsupervised learning problem in which we seek to discover some structures (interesting locations) in an unlabeled dataset. The hidden Markov model (HMM) is a statistical model whose ability to model data without prior knowledge of the hidden states makes it useful in a variety of settings. It is used in this work to model the mobility behaviour of each user, considering the days of the week and areas of interest predicted by the DBSCAN. Thus, these two techniques are combined to understand the mobility behaviour of each user in the dataset.
The term de-anonymization is sometimes referred to as re-identification. Throughout this thesis, both terms may be used interchangeably.

4.2.1 The DBSCAN

The Density-based spatial clustering of applications with noise (DBSCAN) is a well-known algorithm in data mining [EKSX96]. The main idea is to use a distance and points threshold input to group data points closer together as a cluster. It excels at detecting data patterns and can easily differentiate dense clusters from low clusters. Moreover, the DBSCAN can form clusters of varying shapes and is very robust to outliers and noise. Its robustness to outliers makes it applicable to mobility datasets, as they generally have many outliers and noise. The DBSCAN algorithm has two main parameters input:

- Epsilon(eps)- It defines the radius distance between two data points. Thus, the distance between two points less than or equal to *eps* belongs to the same cluster.
- minPoints- This input defines the minimum number of data points that can form a cluster.

After eps and minPoints are determined (supplied by the user), the algorithm

randomly picks a starting point by creating a circle (using *eps*). Following that, a distance measurement is used to determine the distance between two data points (for example, the Euclidean distance). The starting points are then categorized as core points (greater than minPoints), border points (contains at least one data point but less than minPoints) and finally, noise (has zero data points within epsilon distance). The clusters are finally computed using the concept of reachability (within epsilon distance) and connectivity (defined by two points belonging to the same cluster). For a full explanation of DBSCAN, we refer the reader to [EKSX96]

4.2.2 Hidden Markov Model (HMM)

A hidden Markov model is a statistical tool used in modelling sequential observations (visible) that probabilistically depend on a hidden sequence of events (hidden states). Our work considers both the spatial (areas of interest, or AoI) as well as temporal (days of the week) factors in the formation of these clusters. We propose a hidden Markov model [BP66, BS68] to define a user's underlying AoI (e.g. home, place of work or worship etc.) by observing the sequences of days of the week. This allows us to build a mobility profile for each anonymous user to aid us in de-anonymizing them.

4.3 Hidden Markov Model (HMM) for Location Trajectories

In order to model trajectories, in this work, we formally parameterize a hidden Markov model as $HMM = \{N, \Omega, \gamma, \beta, \pi\}$, where:

- $\boldsymbol{N}$, is the number of states $(\triangle)$. The states represent the area of interest (AoI) visited by the user on any particular day as identified by the clustering algorithm (based on the assigned parameters) used. The area of interest (hidden states) is basically where a user frequently visits and spends considerable time. Such places, typically generated using a clustering algorithm, are usually sensitive, and the user may not wish to disclose them. These locations include but are not limited to one's home, work, place of worship, and shopping centre. The place of visits or area of interest together make up the state set; $\triangle = \{\triangle_1, \triangle_2, \cdots, \triangle_N\}$.
- $\boldsymbol{\Omega}$, is the number of distinct symbols observed. For example, the observed symbols in this work are the set of distinct days (δ) of the week. Thus, the

maximum number of Ω is 7. These observations are captured when the user visits any of these states. Hence, if a user visits more than one area of interest (state) on a particular day, say Tuesday, then this day would be recorded multiple times in line with the number of states visited. The sequence of observation symbols, $\delta_1 \cdots \delta_T$, has a maximum size of T.

- $\boldsymbol{\gamma}$, is an $N \times N$ transition probability matrix with members represented as γ_{ij}. It basically represents the probability distribution of transiting from one area of interest (state) to the other. Each member γ_{ij} is the probability of moving from state $\triangle_i$ to state $\triangle_j$, i.e., the number of times a user moves from state $\triangle_i$ to state $\triangle_j$ over the number of times a user moves from state $\triangle_i$. We formally define γ_{ij} as:

$$\gamma_{i,j} = \frac{\sigma(i,j)}{\sum_{j=0}^{N-1} \sigma(i,j)}$$

  where $\sigma(i,j)$ is the number of transitions from state $(\triangle_i)$ to state $\triangle_j$ such that $\sum_j \gamma_{i,j} = 1 \; \forall \; i$.
- $\boldsymbol{\beta}$, is the emission probabilities with members represented as $\beta_{i,j}$, which is an $N \times \Omega$ matrix. It is the probability of a given state $(\triangle)$ generating a particular observation symbol (δ). That is the probability that a user visited an area of interest $(\triangle)$ on a particular day (δ).

$$\beta_{i,j} = \frac{\tau(i,j)}{\sum_{j=0}^{\Omega-1} \tau(i,j)}, \quad \text{where} \quad \begin{matrix} 1 \leq i \leq N \\ 1 \leq j \leq \Omega \end{matrix}$$

  $\tau(i,j)$ is the number of times a user visits an area of interest $(\triangle_i)$ on a given day (δ_j), also $\sum_j \beta_{i,j} = 1 \; \forall \; i$
- $\boldsymbol{\pi}$, is the initial probability distribution over all states $(\triangle)$, where

$$\pi_i = \frac{\sigma_i}{\sum_{j=0}^{N-1} \sigma_j}, \quad \text{and} \quad 1 \leq i \leq N$$

  where σ_i is the total number of times, a user first start from a state $(\triangle_i)$ or a place of interest (AoI) for all given days.

Given the temporal dynamics of mobility patterns of the data, which is particularly relevant in this work, HMM was used in the construction of the user profiles (fingerprinting) due to its effectiveness in modelling the temporal sequence of events or sequential data. This is particularly important, considering the fact that the day of the week significantly impacts people's daily activities. Furthermore, HMM excels in capturing the probabilistic nature of user trajecto-

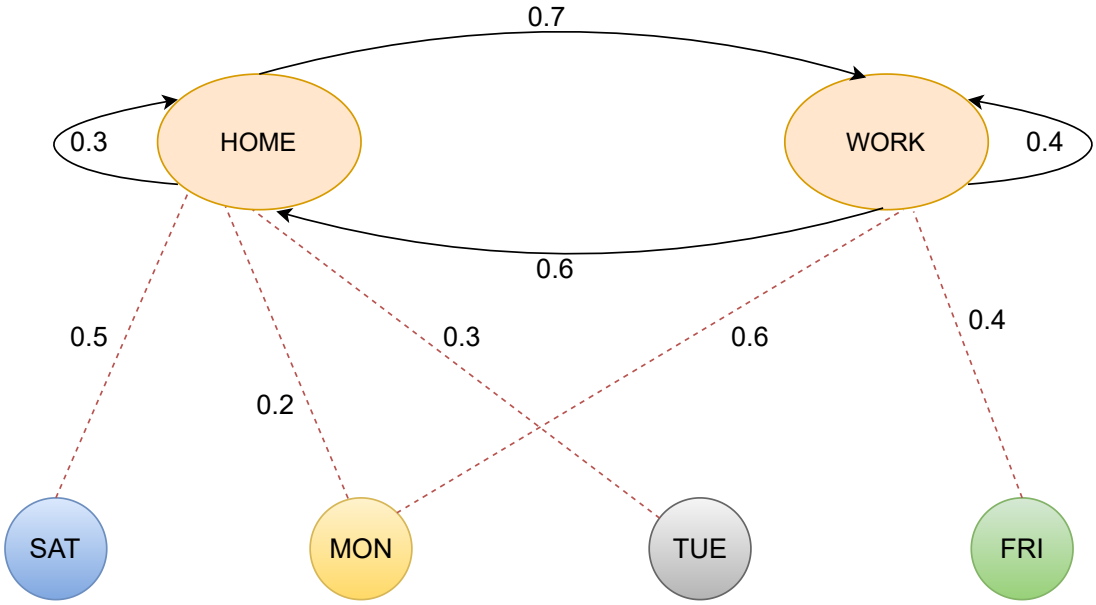


Figure 4.1: Illustrates an Example of a Fully Connected (Ergodic) 2-States Hidden Markov Model. The Model Relates Days of the Week (Observations), Saturday, Monday, Tuesday, and Friday, to Areas of Interest (Home and Work) With Both Transition and Emission Probabilities.

ries and their temporal dependencies. This capability is especially crucial when examining the probabilistic links or relationships between frequently visited locations (areas of interest) to reconstruct user mobility patterns or for user fingerprinting. Thus, the sequential nature of mobility patterns makes HMM more suitable for handling transitions between different locations over time. Such type of modelling is crucial for differentiating users within a target dataset, as each user's mobility pattern can be uniquely represented within a probabilistic framework. Leveraging such a characteristic is vital for the purpose of this work, where the inference of user mobility patterns from sparse or anonymized dataset is not straightforward.

Contrary to conventional approaches, where frequently visited locations might be assigned as observables and days of the week with hidden states, this thesis adopts a reverse methodology. This decision stems from the goal of the de-anonymization, which diverges from predicting the immediate future movements or next locations as done in existing literature. Instead, the goal is to construct the underlying pattern movement that characterize each user. By representing areas of interest as the hidden states and days of the week as observables, our model adeptly captures the uniqueness of user mobility. This is done by factoring the transition probabilities between frequented locations and incorporating the temporal probabilities of visits to these locations, thereby im-

proving the accuracy of the attack as evidenced in section 4.6.2.

4.4 Experimental Setup and Pre-processing

In preparing the target dataset for profiling of users, we sort the dataset by the number of days spent by each user and divide it into two non-overlapping datasets: the training dataset, which accounts for approximately 80% of the total ground-truth dataset, and the auxiliary dataset (testing dataset), which accounts for the remaining 20%. This is in deviation from other setups in the literature [FSH11, MYYR10], where the testing dataset is not disjoint from the training dataset (and in fact is a subset of it). As argued in [GKdPC14], this leads to a testing dataset that is heavily biased, and to artificially high de-anonymization results. Similarly to [GKdPC14], our dataset is instead non-overlapping (trained and testing datasets are disjoint) to avoid bias. Unlike [GKdPC14], however, where the dataset was split into two halves, one for training and one for testing, in this work the former makes up the bigger (80%) portion of the dataset. In a real-world scenario, this would constitute anonymous users' mobility datasets published by a service provider. The testing dataset is the adversary's knowledge of known user mobility traces, which should constitute a small fraction of the published mobility dataset and not necessarily be the same as the published datasets. The assumption is that the adversary may only gain limited knowledge about the mobility behaviour of the target user.
Furthermore, for each user in the anonymous dataset (training datasets), a clustering algorithm is used to group each user's mobility traces (longitude and latitude) into clusters (Area of Interest, or AoI). The number of clusters (AoI) output depends on the clustering algorithm and the temporal and spatial resolution of the raw mobility traces. For instance, selecting smaller values for the selected clustering algorithm parameters may return more areas of interest (states) per user, which, in effect, may impact the success of de-anonymization. These clusters (AoI), generated by the clustering algorithm, act as the hidden states in the HMM as defined in Section 4.3. Generally, these clusters or areas of interest contain sensitive information (like home, place of work, place of worship, the type of health facility the user usually visits, etc.) that the adversary is likely to discover for each anonymous user in the target database (training dataset). Figures 4.3 and 4.4 each show an example of a different user whose mobility trajectories are classified by the clustering algorithm into five clusters (areas), with the black spots indicating noise (outliers). Figures: 4.3 and 4.4

depict users included in the Taxicabs dataset and the GeoLife dataset, respectively, both of which are discussed in Section 4.6.1. The specific days of the week during which users entered these clusters (areas of interest) represent the observational sequence used to learn about these areas of interest (clusters). Then, using the HMM algorithm, we create a profile of each anonymous user's mobility traces (Markov model) from the training data.

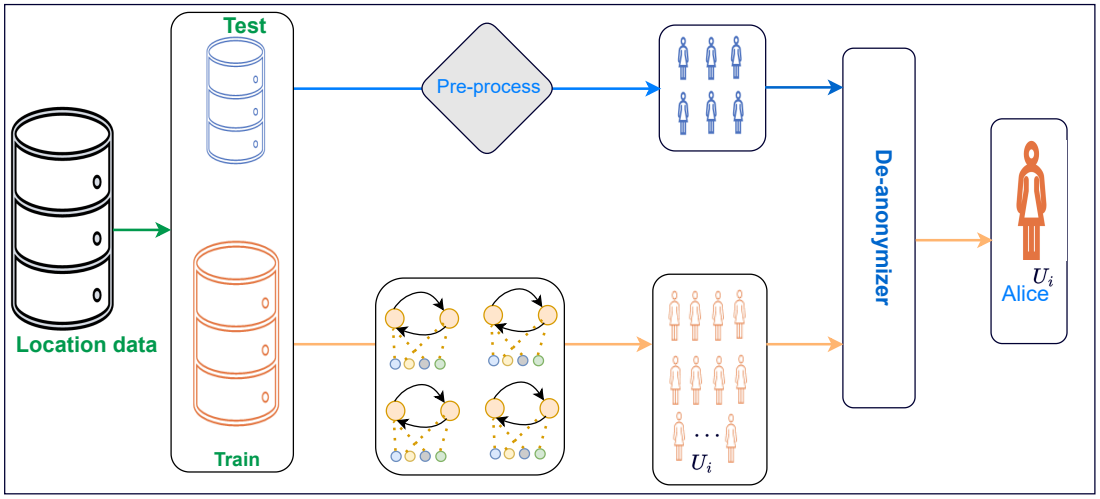


Figure 4.2: An Adversary Model Depicting How the Adversary Successfully Performed the De-anonymization Attack. The Target Dataset Is Split Into Training and Testing. Users' Profiles (Fingerprints) Are Constructed Using HMM From the Training Dataset. Passed to the De-anonymizer Are the Processed Mobility Traces of a Target User From the Testing Dataset and the Lists of Generated Profiles for Successful Reidentification.

The transition probability of 0.7 figure 4.1 indicates the probability of the user going to work from home, and vice-versa for the transition probability of 0.6. The transition probability of 0.3 from home to home could indicate that the user may have left the house to make a quick trip to the store, coop, or even to accompany a friend and quickly return home. Such a place may not be captured as an area of interest if the user visits there infrequently. These profiles act as the fingerprint or unique identifier (trajectory model) for each anonymous user in the target database. An adversary can then easily associate one of these profiles with a target user whose identity and a limited set of spatiotemporal trajectories (testing dataset) are known to the adversary. Figure 4.2 presents a diagrammatic representation of the adversary model of how the process of de-anonymization is carried out. This model is used to understand better how de-anonymization works. This view begins with the generation of the users' profiles from the target dataset (the ground truth mobility dataset) and goes all the way through to the final stage of the reidentification process.

To speed up the re-identification process, we created a database (Λ) to hold the model (λ) of each anonymous user such that: $\Lambda = \{\lambda_j\}_{j=1}^{n}$, where $\lambda_j = \{N, \Omega, \gamma, \beta, \pi\}$ is the model parameters for each anonymous user, and n is the total number of anonymous users. The database (Λ) is updated each time the temporal and spatial resolutions are modified. Depending on the selected resolution parameters, the clustering algorithm may identify more or fewer areas of interest and accordingly affect the overall computation of the model parameters $(N, \Omega, \gamma, \beta, \pi)$ and consequently affect the re-identification process. The effect of this modification is illustrated in the analysis in Section 4.6.2
The testing data, which is just a fraction of the training dataset, is assumed to be the auxiliary knowledge (Spatio-temporal points) the adversary has about the target user and wants to link to one of the anonymous users in the target database (training dataset). In the sense of auxiliary information, these spatiotemporal datasets (testing dataset) are similar in granularity to that of the target database (training dataset) but not a subset of the target database.

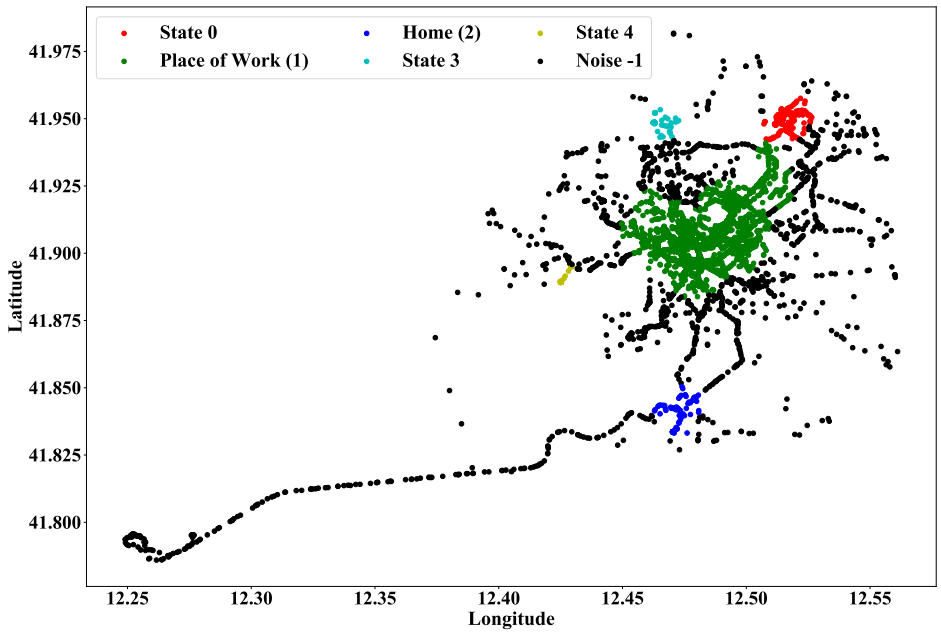


Figure 4.3: Taxicabs: A Taxi Driver With Five Different Areas of Interest (Clusters), With the Black Points Representing Noise. The Spatial Traces Consist of GPS Coordinates Gathered With an Android Tablet in Rome.

Thus, the adversary knows the target user but owns very limited Spatio-temporal trajectories (testing dataset) about this user and wants to link this

auxiliary information to one of the anonymous users in the database. Though these users in the target database are unknown, their Markov mobility patterns containing sensitive information are known to the adversary.
For this analysis, and in order to be able to quantify the success rate, we assume the number of users in both the training and testing datasets is equal. Although this may not be the case in a real-world scenario, where the adversary is likely to have just a fraction of the number of people published in the target dataset (training dataset), this is a common approach in the literature. For this reason, let $\lambda^{\star}$ be the model (built from the auxiliary knowledge) for the target user the adversary knows about. This would be needed in performing the re-identification attack. A preprocessing such as encoding the user's observational sequences ($\delta_1 \cdots \delta_T$) is performed before feeding the testing dataset into the de-anonymizing algorithm. This helps the model to understand better to assign a weight to the dataset correctly. Depending on the selected de-anonymizing algorithm and the dataset in question, the testing data is tuned by modifying the Spatio-temporal resolutions (in some cases, a new model is built on the testing data) before performing the re-identification attack.

4.5 De-Anonymizers

In this section, two de-anonymization attack techniques (de-anonymizers) are presented: the *Forward* algorithm and a *Kullback-Leibler divergence*-based algorithm. The Forward algorithm is presented in Section 4.5.2. As it relies on the likelihood estimator by generating sequence of conditional probabilities in computing the likelihood, we also present the likelihood estimator in Section 4.5.1. The Kullback-Leibler divergence-based algorithm (Section 4.5.3), on the other hand, relies on the divergence between the two probabilistic distributions by quantifying their distance.
This work employs two different types of algorithms to comprehensively analyze the performance of the de-anonymization process under different varying conditions based on the target users.Particularly highlighting the differential methodologies employed in the re-identification process. For example, the Forward de-anonymizer is more focused on conditional probabilities across all paths, whereas the Kullback-Leibler divergence is more focused on quantifying the dissimilarity or relative entropy between probability distributions as mentioned earlier. This comparative analysis allows for a more in-depth understanding of the impact of variables other than the observational sequence,

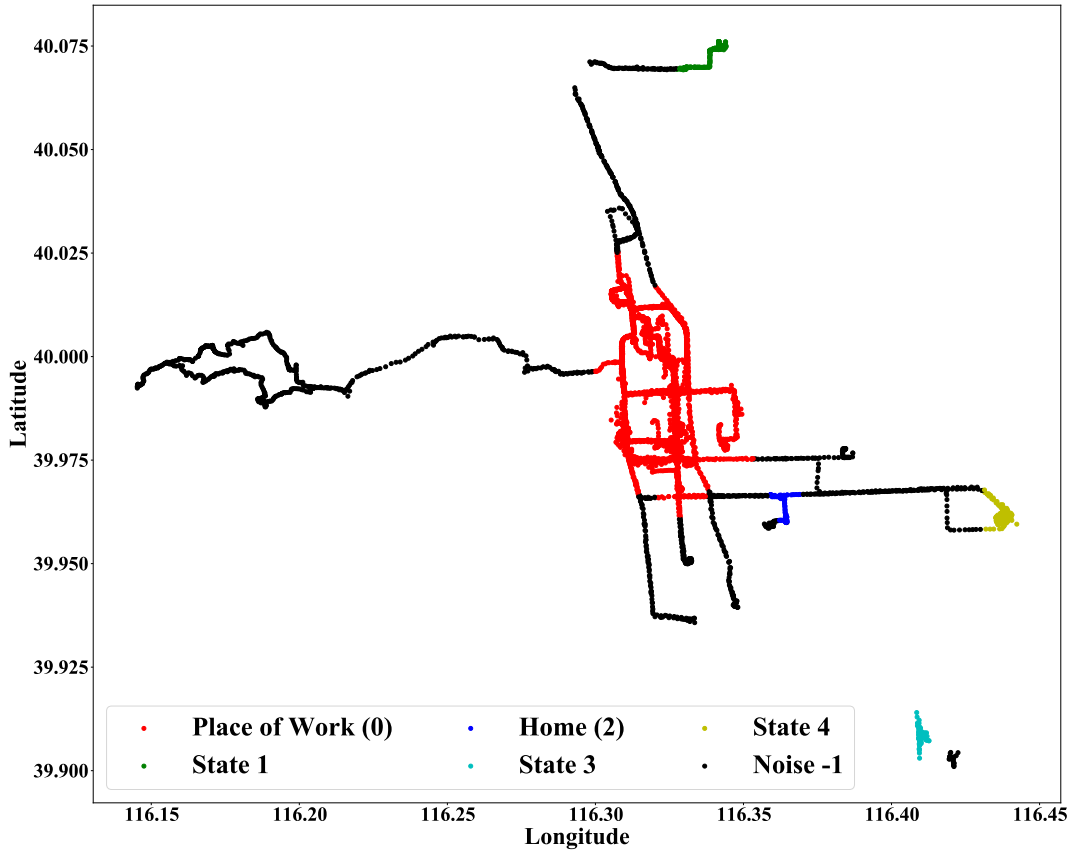


Figure 4.4: The Spatial Representation of a User's Movements in the GeoLife Dataset: 5 Areas of Interest (Clusters) Are Recognized, With Black Points Being Noise. Where an Area Is Not Connected to the Rest of the Graph, This Could Indicate That the Location Is Reached Via Means That Prevent a GPS Signal From Being Received (For Instance, Underground Metro or a Tunnel).

such as the time they are generated, on the final prediction of a target user. Furthermore, the utilization of both algorithms facilitates how uniquely users' profiles are exposed to both algorithms, that is, how accurately can both algorithms accurately predict the targeted users correctly. This cross-validation approach between the anonymizers aims to ensure a robust analysis of the re-identification while also revealing the strengths and limitations of each in the context of the re-identification accuracy process. The results, as illustrated in section 4.6.2 underscore the differential impact of various factors on the prediction capabilities of each algorithm. Building on this analysis, the practical application of these techniques on real-world datasets further illustrate the po-

tential of the re-identification process. Experimental result, discussed in Section 4.6, showcase the effectiveness of both algorithms in re-identifying users within the target dataset

4.5.1 Likelihood

One of the problems a hidden Markov model can solve in a real-world scenario is the probability for an observation sequence of size T (assumed to be generated by N-states) to belong to the model, or for the model results in such an observation sequence, i.e., $P(\delta|\lambda)$. The intuition behind the likelihood score is that if a model (λ) is built based on observation sequence; $\delta = \delta_1 \cdots \delta_T$ (from the target database) similar to the one created by an individual (target user), this should result in a higher likelihood score, $P(\delta|\lambda)$. Ideally, computing the probability of such an observation sequence $\delta = \delta_1 \cdots \delta_T$ can occur when we know the states sequence $\triangle = \triangle_1 \cdots \triangle_T$. Thus computing the probability of the observational sequence considering all possible states sequence is given by:

$$\begin{aligned} P(\delta|\triangle, \lambda) &= \prod_{t=1}^{T} P(\delta_t|\triangle_t, \lambda) \\ &= \beta_1(\delta_1) \cdot \beta_2(\delta_2) \cdots \beta_T(\delta_T) \end{aligned} \tag{4.1}$$

Moreover, the probability of states sequence given the model is:

$$P(\triangle|\lambda) = \pi_1 \cdot \gamma_{12} \cdot \gamma_{21} \cdots \gamma_{T-1T} \tag{4.2}$$

Hence the joint probability of being in a state $(\triangle)$ of a specific type of observation is given by:

$$P(\delta, \triangle) = P(\delta|\triangle)P(\triangle) \tag{4.3}$$

Therefore, the probability of the observation sequence assumed to be generated by the T states sequence is by summing the joint probability over all possible hidden states:

$$P(\delta) = \sum_{\triangle_T} P(\delta, \triangle) = \sum_{\triangle_T} P(\delta|\triangle)P(\triangle) \tag{4.4}$$

With the given model λ, the observational sequence probability becomes:

$$\begin{aligned} P(\delta|\lambda) &= \sum_{\triangle_T} P(\delta|\triangle, \lambda) P(\triangle|\lambda) \\ &= \sum_{\triangle_T} \pi_1 \beta_1(\delta_1) \cdot \gamma_{1,2} \beta_2(\delta_2) \cdot \cdots \gamma_{T-1,T} \beta_T(\delta_T) \end{aligned} \tag{4.5}$$

A direct computation of equation (4.5) for N states and T observations is computationally infeasible due to N^T possible states sequence, especially when both N and T are large numbers. Thus having a computational complexity of approx $2T \cdot N^T$ for such computations is not ideal in calculating the likelihood. A more reliable and efficient algorithm to avoid such an exponential algorithm is the *Forward* algorithm [BS68, Rab89]. The derivation is similar to the one presented in [Rab89].

4.5.2 The Forward De-anonymizer

The brute-force summation procedure for every possible state is not feasible in practice. Thus the use of the *Forward* algorithm [BS68, Rab89] to make such computations. Like dynamic programming, the *Forward* algorithm reduces the number of calculations when computing the observational probability. It computes the observational probability by summing the probabilities for all paths (hidden states), resulting in such an observational sequence.
To compute the *Forward* algorithm, let's define a forward probability (*forward parameter*); $\alpha_t(i) = P(\delta_1 \cdots \delta_t, \triangle_t = i|\lambda)$, which is the probability of the observational sequence, $\delta_1 \cdots \delta_t$ after being in state $\triangle_i$ at time t for the given model (λ).
In computing the helper vector (forward parameter) inductively: First step:

$$\alpha_1(i) = \pi_i \cdot \beta_i(\delta_1) \quad \text{where} \quad 1 \leq i \leq N \ , \tag{4.6}$$

Inducting step gives:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) \cdot \gamma_{ij} \right] \cdot \beta_j(\delta_{t+1}) \ , \tag{4.7}$$

where

$$\begin{aligned} 1 \leq j \leq N \\ 1 \leq t \leq T-1 \end{aligned} \ ,$$

and where:

- $\alpha_t(i)$ is the probability of observing $\delta_1 \cdots \delta_t$ up to being in state $\triangle_i$ at time t ,
- γ_{ij} is the transition probability of moving from state $\triangle_i$ to state $\triangle_j$,
- $\beta_j(\delta_{t+1})$ is the probability of observing δ_{t+1} at state $\triangle_j$.

Finally, summing all $\alpha_T(i)'s$ at final time step $t = T$ for all possible previous states gives:

$$P(\delta|\lambda) = \sum_{i=1}^{n} \alpha_T(i) \tag{4.8}$$

Comparing this computation to the previous brute-force summation of the likelihood for all possible states (4.5.1) reduces the computation to order $O(N^2T)$, where N^2 is as a result of computing N previous states (N elements of α_t) to all N states for each T observations. Thus, with this forward computation, we can generate a score for each model (anonymous users whose mobility traces are known) to aid us in the re-identification (de-anonymization) process.
We explore this solution by building a scoring algorithm (maximum likelihood score) to de-anonymize (re-identify) the anonymous users (whose mobility traces are known) from the training dataset with a certain threshold. Computing such a score, all the anonymous users' models in the target database (Λ) are each called for each target user's observational sequence and the results (the likelihood score) are stored in the form of a matrix for the re-identification analysis. Thus, let $\{\tilde{\Omega}_i\}_{i=1}^{n}$ be the total target users with a known observational sequence in the testing dataset and n be the total number of users where $\tilde{\Omega}_i = \delta_1^i \cdots \delta_T^i$ is the observational sequence for target user i. Hence, from equation (4.8), we compute $P(\tilde{\Omega}_i|\lambda_j)$, the likelihood scores for user i from the testing dataset given models λ_j from the training dataset (anonymous users). Thus, from the likelihood scores, the model λ_j (an anonymous user) that generates the maximum likelihood score is likely to be the corresponding user from the testing dataset (known users).

4.5.3 Kullback-Leibler Divergence De-anonymizer

The next de-anonymization algorithm relies on a divergence measure or relative entropy based on the Kullback-Leibler divergence [KL51], which in itself is a generalization of Shannon's entropy. This divergence measure was first proposed by [JR85] as a distance measure for any two Markov models. It measures the difference between the two models' log probabilities based on the observations generated by one of the models. Technically, it's not a distance

measure as it does not satisfy the symmetric or triangular inequality property. The divergence measure $\Gamma(\cdot,\cdot)$, between two models λ_1 and λ_2 is defined as:

$$\Gamma(\lambda_1,\lambda_2) = \frac{1}{T}[log\ P(\delta^{(2)}|\lambda_1) - log\ P(\delta^{(2)}|\lambda_2)] \tag{4.9}$$

where $\delta^{(2)} = \delta_1 \cdots \delta_T$, is the observation sequence of size T generated by the model λ_2. Looking at equation 4.9, one can see that it is just the difference in the likelihood computed in equation 4.8 between the models $\lambda_1\ and\ \lambda_2$. The interpretation of equation 4.9 is how well is model λ_1 different from model λ_2 or how divergent is model λ_1 from λ_2 based on the observation sequence generated by one of the models.
As this score measure is not symmetric, we employ the Jensen-Shannon divergence (JS) to make it symmetrized and smoothened. The Jensen-Shannon divergence (JS) between models λ_1 and λ_2 is given by:

$$JS(\lambda_1||\lambda_2) = \frac{1}{2}\Big[\Gamma\Big(\lambda_1, \frac{(\lambda_1+\lambda_2)}{2}\Big) + \Gamma\Big(\lambda_2, \frac{(\lambda_2+\lambda_1)}{2}\Big)\Big] \tag{4.10}$$

Based on this divergent measure, we are able to build an evaluation score for all the trained models to make predictions. A divergent score between λ_1 and λ_2 closer to 0 simply means the two models, λ_1 and λ_2, are likely to be identical. The adversary explores this algorithm by constructing a probability model of the target user's Spatio-temporal trajectories, similar to the one built for the users in the target database. This model comprises transition probabilities (the likelihood of moving from one area of interest to the other) and emission probabilities representing the days the users were seen in these locations (areas of interest). Consequently, for each target user ($\lambda^\star$), the adversary computes the divergence measure between $(\lambda^\star)$ and each anonymous user in the target database, $\Lambda = \{\lambda_j\}_{j=1}^{n}$. From the divergent scores between the target user and the anonymous users, the pair models (target user and any of the anonymous users) that result in the minimum score; $\min(\lambda^\star, \lambda)$ for a given threshold indicates a corresponding match in the target database (Λ), i.e., the two models are likely to be identical.

4.6 Experimental Evaluation

We evaluate the performance of our de-anonymization attacks based on individual-level ground truth mobility trajectories from two different groups of

users. The two proposed de-anonymization attacks are both based on the target user's blueprint (unique Spatio-temporal profile) built using the HMM. Parameters such as spatial and temporal values were varied to test the robustness of the proposed de-anonymizers. In addition, the success of the de-anonymizers was compared with the state-of-the-art de-anonymizers in the literature. Specifically, we compare our work to other attacks that used a similar dataset to compute the accuracy of their algorithms.

4.6.1 Dataset

In evaluating the efficiency of the model and the de-anonymization attacks, we exploit publicly available independent real-life mobility traces from two different groups of users in two different cities: Shanghai (GeoLife) [ZZXM09] and Rome (taxicabs) [BBL$^+$].

The GeoLife [ZZXM09] dataset, consisting of 182 users' GPS trajectories, was collected by Microsoft Research Asia spanning from April 2007 to August 2012 in Shanghai. The dataset collected at a high rate of 1 to 5 s covers users' daily routines such as shopping, sporting activities, workplace, going home, etc. Users whose mobility traces covered less than eight days of GPS trajectories were filtered out for this evaluation.

However, the taxicabs [BBL$^+$] dataset captures mobility traces of taxi cabs in Rome, Italy using GPS devices. The dataset contains GPS trajectories of approximately 320 taxis collected over 30 days (from 1 February to 2 March 2014) at a high temporal resolution of about a second. Unlike the GeoLife dataset, which has the trajectories of each user in a separate folder, the taxicab dataset has all users (taxi drivers) in a single file. This required additional pre-processing to sort the trajectories according to the IDs assigned to each driver. A peculiarity of the taxicab dataset is that it represents trajectories of users whose work is not carried out in a static location but rather in motion (normally within the confines of a specific area of the city). This is similar to other professions and jobs, including delivery drivers, sales representatives etc. It is important to note that the dataset contains all user movements, including those to and from home. In this respect, it is similar to other categories of users.

The ability of a model to successfully re-identify users in both datasets, despite the variations between them, indicates flexibility in the attack.

4.6.2 Experimental Results

Our experiment considers how the de-anonymizers perform separately on the two different datasets (GeoLife [ZZXM09] and taxicabs [BBL+]) and how the performance rates are affected by varying spatial resolution. As presented in figure 4.5, the success rate of the *Forward* de-anonymizer (D_{FW}) on the GeoLife dataset (shown in the colour green) ranges from 72% to 65%, depending on the chosen radius distance. This performance is a significant improvement when compared to the performance of other de-anonymizers in the literature (as discussed in Section 4.6.3).

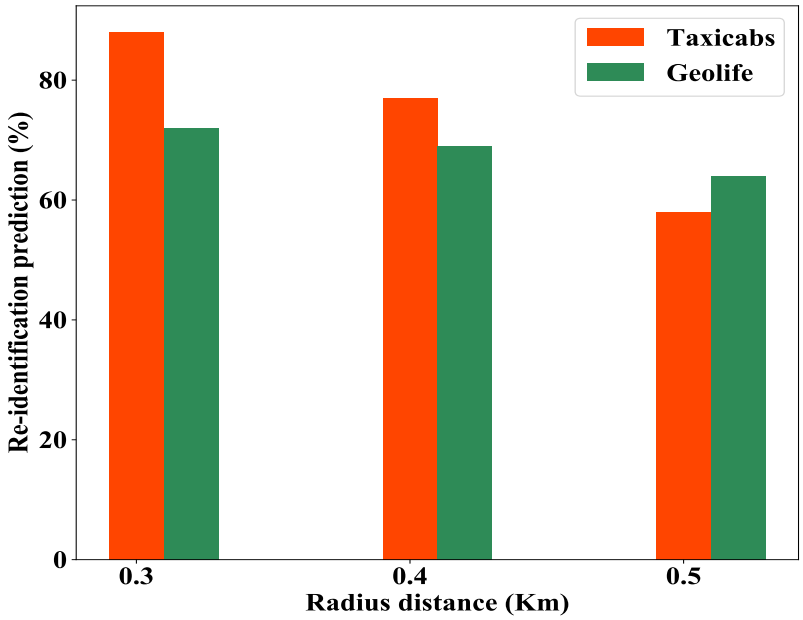


Figure 4.5: The Rate of Performance of the *Forward* De-anonymizer on the GeoLife [ZZXM09] and Taxicabs Datasets [BBL+].

The performance of the de-anonymizer was validated using the taxicabs dataset (represented in orange) from Milan, which had a larger number of users (over 300 users involved). The figure shows that the prediction rate increases sharply from 58% for half a kilometre radius to about 88% as the radius distance decreases. From the figure, the radius distance is inversely proportional to the success rate of the *Forward* de-anonymizer.
Successful re-identification of an anonymous user in the target database implies that there is a similarity in observational mobility trajectories used to build the model and that of the known user, resulting in the algorithm returning the

highest score.

The D_{KL} de-anonymizer was also tested on GeoLife datasets, with even stronger results. Figure 4.6 depicts the success rates of the D_{KL} de-anonymizer on the GeoLife; from the figure, the prediction rate increases from 74% to 85% as the radius distance decreases. When tested on the taxicabs dataset, the D_{KL} algorithm's impressive performance becomes even more apparent. Its performance increases sharply from 62% to a 90%, as depicted in Figure 4.6.

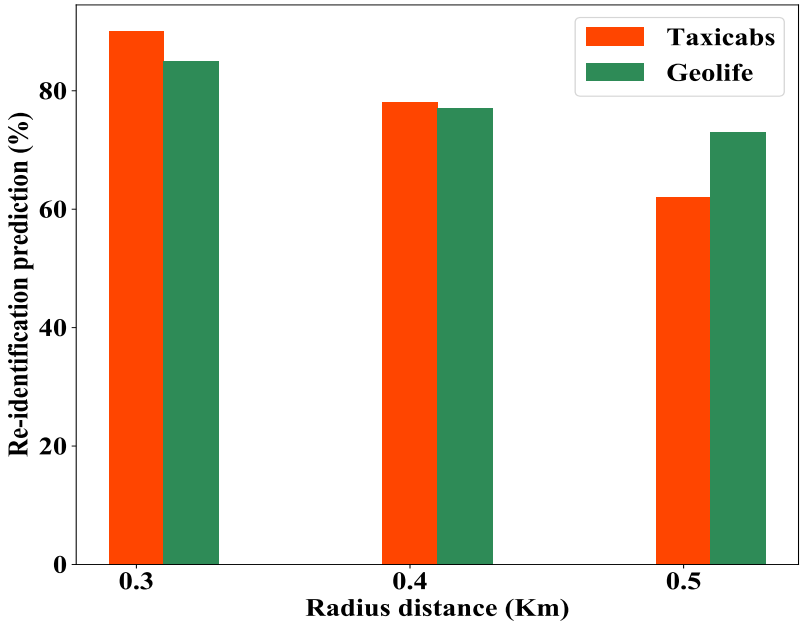


Figure 4.6: The Rate of Performance of the *KL* De-anonymizer on the GeoLife and [ZZXM09] Taxicabs Datasets [BBL$^+$].

Tables 4.2 and 4.3, provide compelling results of the exceptional accuracy demonstrated by our two proposed models in de-anonymizing users in the dataset. Consequently, when either model succeeds in de-anonymizing a user, the results can be trusted with a high level of certainty. Furthermore, both models are proficient in identifying a substantial number of anonymized users within a given dataset, demonstrating their effectiveness and reliability in performing de-anonymization tasks.

When it comes to the performance of both de-anonymizers, the D_{KL} de-anonymizer has a higher prediction rate than the *Forward* algorithm. Figure 4.7 and 4.8 depicts the superiority of the D_{KL} algorithm over that of the *Forward* algorithm (D_{FW}) with varying spatial resolution. In the GeoLife dataset, this difference in performance is even more pronounced. However, the rates

of increase within both de-anonymizers were almost the same as the radius distance decreases. This presupposes that the characteristics of the dataset in question have an impact on the performances of the de-anonymizer.

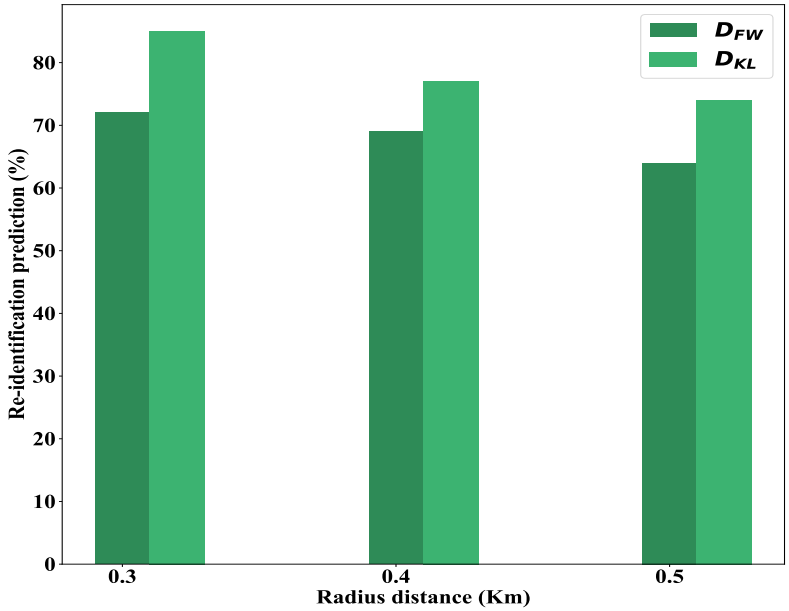


Figure 4.7: Performance Rate of the De-anonymizers on the GeoLife Dataset [ZZXM09].

A property that makes the D_{KL} de-anonymizer more effective is the observational property matrix (emission probabilities). Deviations in β return a higher dissimilarity score D_{KL} than in γ (transition between areas of interest). Consequently, users are more likely to be re-identified if they normally visit areas of interest on particular days. Naturally users gravitate to places of interest on specific days, like going to the theatre on Wednesdays or to a stadium on Saturdays. For example, consider two users who usually visit the same location, say a shopping mall; the algorithm easily re-identifies them in a target database as they normally make this visit on different days. Thus even if the trajectories are similar, they are still likely to be re-identified because of the weight placed on the identity matrix.

The superiority of D_{KL} de- anonymizer's performance is underscored by its consistently higher $F1$ Scores across Tables 4.2 and 4.3 at every resolution level. This demonstrates a more balanced and effective trade-off between Precision and Recall scores in the GeoLife [ZZXM09] and Taxicabs [BBL$^+$].

Another noteworthy finding in Figures 4.7 and 4.8 is that, while both de-

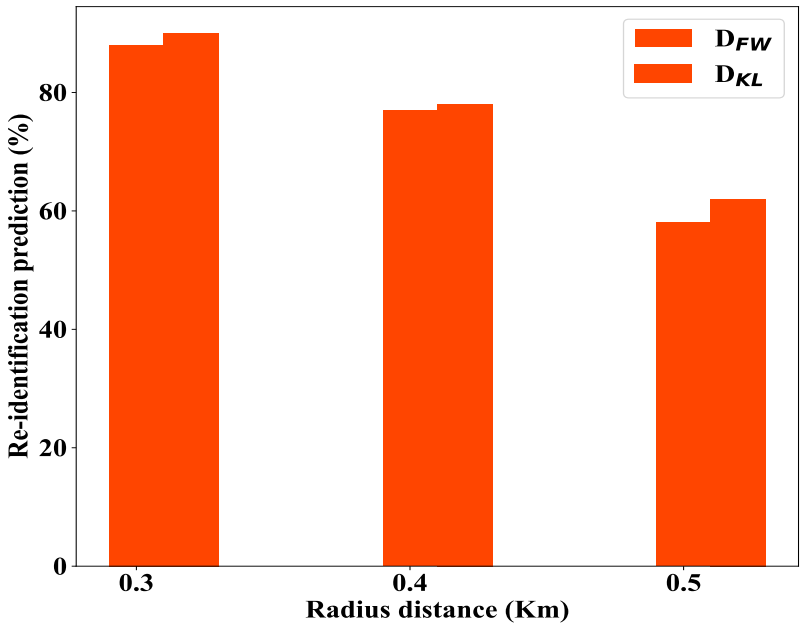


Figure 4.8: Performance Rate of the De-anonymizers on the Taxicabs [BBL+].

Table 4.2: Performance Comparison of Model D_{KL} and Model D_{FW} on GeoLife Dataset Measuring Precision, Recall, and F1 Scores Up to Three Decimal Places

	GeoLife Dataset					
	Model D_{KL}			**Model D_{FW}**		
Resolution (km)	**0.3**	**0.4**	**0.5**	**0.3**	**0.4**	**0.5**
Precision Score	0.932	0.867	0.867	0.907	0.907	0.892
Recall Score	0.841	0.761	0.725	0.716	0.682	0.636
F1 Score	0.792	0.710	0.669	0.643	0.606	0.551

Table 4.3: Performance Comparison of Model D_{KL} and Model D_{FW} on Taxicab Dataset Measuring Precision, Recall, and F1 Scores Up to Three Decimal Places.

	Taxi Dataset					
	Model D_{KL}			**Model D_{FW}**		
Resolution (km)	**0.3**	**0.4**	**0.5**	**0.3**	**0.4**	**0.5**
Precision Score	0.948	0.878	0.794	0.954	0.907	0.851
Recall Score	0.899	0.779	0.614	0.876	0.772	0.577
F1 Score	0.872	0.741	0.548	0.843	0.712	0.478

anonymizers appear to perform better in the taxicab dataset (figure 4.8) than in the Beijing dataset (Figure 4.7), as the radius distance increases, both de-anonymizers in Figure 4.7 of GeoLife dataset perform significantly better than in the taxicab dataset (Figure 4.8). In fact, both de-anonymizers have a higher prediction in the GeoLife dataset than in the taxicabs dataset at half a kilometre radius. This is even more evident looking at Figure 4.5 and Tables 4.2 and 4.3. This performance can be attributed to the fact that users in the GeoLife dataset seem to have more varying mobility trajectories in wider areas, making them easily identifiable (due to higher unique features per user). However, the same cannot be said for taxicab drivers who are likely to have more similar mobility patterns in wider areas. Taxicab drivers, for example, are likely to use a famous street or a road network connecting to a popular location, such as a stadium or an airport. For instance, most of the drivers in taxicabs [BBL$^+$] were using the *A91* highway to the Leonardo da Vinci International Airport. In contrast, users are likely to have many different routes to these same locations. As a result, identifying users with similar mobility patterns is more complicated than identifying users with varying mobility patterns.

From the analysis, nearly 66% of the user profiles (Spatio-temporal user patterns) in the GeoLife dataset built by the HMM were so distinct that both the *Forward* and D_{KL} de-anonymizers could re-identify these same individuals. As a result of these users' unique mobility patterns based on the HMM model, the attacker could re-identify them regardless of the de-anonymizer used in this work. Furthermore, one intriguing finding from the study is that when both de-anonymizers predict the same user, as in 66% of the cases in the GeoLife dataset, then it is almost a certainty. At no point did the two algorithms (*Forward* & D_{KL}) predicted the same user, which turned out to be false. This gives the assurance that combining these two algorithms in the de-anonymization attacks will yield a result with a high degree of accuracy.

4.6.3 Comparison to Previous Attacks

As the GeoLife dataset [ZZXM09] is frequently used in the literature, it is commonly regarded as a benchmark. For this reason, we compare our results to previous results on the basis of this dataset.

To make a fair comparison between our work and other known de-anonymization attacks, we only consider previous works that used non-overlapping training and the testing dataset. As discussed in Section 4.4, and to avoid ambiguity, the training and the testing dataset should not be

overlapping but rather disjoint (non-overlapping) to avoid bias. As evidenced in [GKdPC14], such bias results in artificially high success rates for de-anonymization.
The works of MMC [GKdPC14], and UHMM [WZF+15] best met the criteria outlined above. Both schemes are briefly outlined in Section 4.7.
Figure 4.9 shows how our algorithm compares to MMC [GKdPC14], and UHMM [WZF+15] in terms of success rate. As illustrated in Figure 4.9, our de-anonymization attack outperforms other algorithms when using the same parameters; our attacks D_{FW} and D_{KL} succeed at a rate of 72% and 88%, respectively, compared to 40 to 45 per cent for UHMM [WZF+15] and MMC [GKdPC14], demonstrating the effectiveness of our attack tools.
Further proof of the effectiveness of the proposed Forward and KL algorithms is that they surpass even attacks evaluated over biased testing sets. Specifically, UHMM's achieved attack success rate is still lower than that of our algorithms even when the training and testing datasets overlapped (were not disjoint), at approximately 70%. We, however, discarded this result from the comparison for the reasons outlined above. Similarly, [MYYR10], which achieves a performance rate of up to around 90% based on the cabs dataset in San Francisco, comes closest to our algorithm's performance but was again heavily biased because the testing dataset was directly extracted from the training dataset.

4.7 Related Work

De-anonymization attacks in the context of a geo-located dataset have been in the study for a while. However, prior research has demonstrated that deleting personal identifiers like names, and email addresses etc., is insufficient to anonymize individuals in the target mobility dataset. According to the work of Mulder et al. [MDBP08], individual behaviour in a Global System for Mobile Communication (GSM) network could be used to re-identify users. They employed two different methods of identification: sequence of cells ID and the Markovian model. The latter used static cells (GSM cells) as the states of the model, which made it almost unlikely to form an ergodic (every state could be connected in a single step) model, i.e., the transition between cells is unlikely to happen without neighbouring cells. Gambs et al. [GKdPC14] also built a Mobility Markov Chain (MMC) to de-anonymize users in a mobility dataset. In their work, a state represents each point of interest (frequently visited locations). As with our work, each MMC is associated with a user, which represents

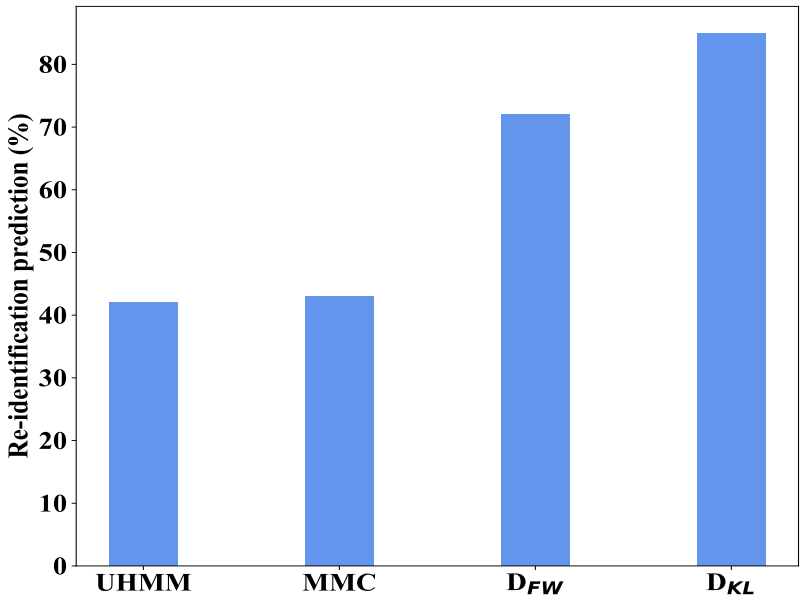


Figure 4.9: Comparing the Performance of Our Method and That of MMC [GKdPC14] and UHMM [WZF+15] on the GeoLife [ZZXM09] Dataset

mobility behaviour. In addition, they proposed distance metrics to measure the similarity between two MMC where the minimum distance is the re-identified user. One of the datasets used to evaluate the performance of their algorithm is the GeoLife dataset. Freudiger et al. [FSH11] worked on re-identification attacks in the context of geolocation. Their work considers both spatial and temporal influences for de-anonymizing the target users. In particular, they create a pair, home/work as an area of interest, where a home is a location where the user typically stays between 9 p.m. and 9 a.m., and work is a location where the user typically stays between 9 a.m. and 5 p.m. The home/work pair is then used as a quasi-identifier to perform the de-anonymization attacks. Wang et al. [WZF+15] proposed a user-hidden Markov model that took spatial and temporal influences into account when de-anonymizing targeted users in a geo-located dataset. Each HMM constitutes the mobility behaviour of the user in the target dataset. Their technique is similar to our hidden Markov model but with a number of differences. First, they divided a day into a 24-time span, which corresponds to the set of states, while a place of visit is regarded as the observational sequence. This is in sharp contrast to our approach, which uses frequently visited locations (AoI) as the set of states and the days of the week as the observational sequence. In addition, their attack was evaluated based on

the GeoLife dataset using the ranking and voting step as the output metric.

4.8 Conclusion

Chapter 4 introduced a highly effective de-anonymization model for geolocation data, leveraging the hidden Markov model to construct user mobility profiles based on Spatio-temporal trajectories. By accurately predicting users' areas of interest, our innovative approach establishes a robust HMM to model users' mobility patterns. The attack techniques demonstrate remarkable success on ground truth mobility trajectories from two different cities, Shanghai and Rome, re-identifying up to 85% and 90% of anonymized users in the GeoLife and Taxicabs datasets, respectively. Our proposed algorithms outshine existing de-anonymization attack methods in the literature, showcasing their potential for real-world applications. The main conclusions of this chapter and others are presented in Chapter 7.1.1 at the end of the thesis.

Chapter 5

A Privacy-Preserving Protocol for Indoor Wi-Fi Localization

In this chapter, we present a well-designed privacy-preserving protocol for indoor Wi-Fi localization, effectively addressing the privacy concerns of users and service providers in location-based services. The research findings described in this chapter were presented in Alghero, Italy and included in the Proceedings of the 16th ACM International Conference on Computing Frontiers (CF '19), hosted by the Association for Computing Machinery in New York, NY, USA.

5.1 Introduction

The prior chapter centred on how de-anonymization attacks can reveal sensitive information, such as the place of worship and an individual's health status, based on the clinics the user normally visits in a location dataset. In addition, and perhaps most significantly, to what extent this sensitive information can be disclosed? The threats to users' privacy that these de-anonymization attacks present can be addressed in a number of different ways. A few of these fall under the category of privacy-enhancing technologies (PETs), such as k-anonymity or secure multi-party computation (SMC). In the context of this thesis, the latter, which is SMC, will be the potential solution to address the challenge of protecting users' privacy in the mobility dataset. The use of SMC: allows the parties involved in the execution of a protocol to jointly compute functions on their private input (like location data points or identities) without revealing such input to the other party. This will preserve the privacy of the individuals whose data is being analyzed. It is particularly useful when dealing with parties with competing interests or mistrusting each other. For example, when two ser-

vice providers need to collaborate on a joint project but are unwilling to share sensitive information about their users or due to competitive advantage, the use of SMC can be useful for them to work together without disclosing any sensitive information in order to avoid compromising their competitive advantage.
Therefore, a protocol that protects users' privacy, such as the one that is proposed in this thesis, is essential in order to address some of the privacy concerns that have been raised as a result of de-anonymization and the ethical use of location data. This is especially true in indoor environments where GPS may not be available or may have a weak signal. In a scenario like this, the Wi-Fi-fingerprinting-based localization approach [BP00, YA05, CFFM15] is currently regarded as the most promising technique for providing accurate information for indoor areas. In particular, a major advantage of this method is that the existing Wi-Fi infrastructure, based on the IEEE 802.11 standard [IEE16], makes it easier and cheaper to deploy with no additional hardware cost.
In order to infer a user's location, Wi-Fi fingerprinting requires a location fingerprinting database known as a *radio map*, built by recording Received Signal Strength (RSS) from various access points (APs) in different locations [KK04, LTK11]. Therefore, fingerprinting localization has two phases: the offline and the online phase. In the offline phase, the Wi-Fi RSS from all the various APs of known positions in the building or desired indoor area is recorded in order to build a database (radio map) saved on the service provider's server. The online phase, where the user interacts with the service, comes with two sub-phases. First, the user measures the location fingerprint (RSS) from all the APs in range at its current location (a grid point). The location fingerprint is then sent to the central server to make a computation which compares the RSS received to the reference radio map built in the offline phase, with the help of a trained decision, in order to determine the location of the user in the building. The computation of the location can be done, among others, using Euclidean distance, likelihood estimator, or support vector machine for regression [BB05]. Although Wi-Fi-fingerprinting localization is promising, unlike GNSS, it can violate user's location privacy or leak user's location information [MTC08, STBH11, LSZ+14]. This is due to the fact that the user computes the location interactively with the server, therefore revealing their location to the service provider. One of the proposed solutions is to allow the client to have full access to the radio map to compute the position locally [YA05, LCC+05]. However, this approach burdens not only the already constrained computational resources of the user (especially in mobile settings) but can also lead to a serious

privacy breach of the service providers's database. In particular, this could leak the entire map of a particular building. If we consider sensitive settings such as an airport, this is clearly not desirable. To solve this problem, most of the literature focus on the privacy of the user neglecting the Service Provider (SP) [YA05, LCC+05], thereby exposing the SP's database. Another area that has not been fully addressed so far is how the SP can have real-time access to the user's location without at the same time violating the user's privacy. For example, an employer may wish to monitor his employees in areas deemed sensitive without violating their privacy elsewhere in the building; and this can be achieved indoors with a secured database that cannot be used for malicious activities by users.

5.1.1 Contribution

In this chapter, we present an indoor location privacy-preserving protocol that allows a service provider to query the location of a user without violating their location privacy. The protocol protects both the user's and the SP's privacy but still delivers the service based on the user's location. Thus, both the SP and the client are mutually distrusting, hence a typical problem of secure multi-party computation. The protocol builds on the data structure presented by Palmieri *et al.* [PCM14, CPM15, CPM18] to protect users' privacy. The protocol also uses homomorphic encryption and the (spatial) Bloom filter (SBF) [PCM14, CPM15, CPM18], a modification of the classical Bloom filter [Blo70], to build trust among both parties.

In particular, we summarize the main contributions of this chapter below:

- To the best of our knowledge, the proposed scheme is the first to use a probabilistic data structure (SBF, a variant of the original Bloom filter) to address the privacy of indoor localization based on Wi-Fi fingerprinting. While other privacy-preserving protocols for indoor localization exist in the literature (such as [LSZ+14], later expanded in [YJ18]), this work does not focus on how the user can obscure the location from the service provider, but on a broader perspective: how a service provider in a private manner can have real-time access to users location without violating their privacy.
- The proposed construction includes an efficient decision algorithm over additive homomorphic encryption that can locate the user with a high probability without leaking any location information, achieving private indoor localization over Wi-Fi fingerprinting.

Table 5.1: Notation: Acronyms and Symbols.

Symbol	Meaning
RSS	Received signal strength
$\boldsymbol{r_j}$	RSS value of an access point j
$\boldsymbol{V_u}$	RSS values of user(u)
V	RSS values for all n AP
C	Encrypted RSS values
B $= \{\triangle_i\}_{i=1}^{s}$	The set of area of interest
SBF	The Spatial Bloom filter
$\boldsymbol{B^{\#}}$	SBF over the set of areas B
$\triangle_i$	An area of interest
$\mathsf{Enc}(\boldsymbol{B^{\#}})$	Encryption of $B^{\#}$ using pk_s
$\boldsymbol{pk_u}$	Public key of user U
$\boldsymbol{sk_u}$	Secrete key of user U
$\boldsymbol{pk_s}$	Public key of server
$\boldsymbol{sk_s}$	Secrete key of server
$\mathsf{Enc}(\mathbf{pk_u}, \boldsymbol{V_u})$	Encryption of V_u using pk_u
$\boldsymbol{\{(\theta_i)\}_{i=1}^{k}}$	The k nearest neighbours
$\{\mathsf{Enc}(\theta_i)\}_{i=1}^{k}$	Encryption of KNN
$\boldsymbol{d}$	User exact location coordinate
$\boldsymbol{B_u}$	SBF build over d
$\boldsymbol{B_u^{\#}}$	Output of $\mathsf{Enc}(B^{\#})$ and B_u

- Further to this, the partial homomorphic computation (based on Paillier's cryptosystem [Pai99] or any other scheme providing additive homomorphism) used to encrypt location information enables privacy-preserving location-based applications based on the private indoor localization. In particular, it guarantees the privacy of both the service provider and the user, such that when the user is not in a sensitive area, the SP learns nothing about the location.
- Finally, the proposed scheme only utilizes the existing Wi-Fi infrastructure and doesn't require new devices for the indoor environment or additional resources.

5.2 Private Indoor Wi-Fi Localization

In this section, we give a general overview of Wi-Fi-fingerprinting. The process can be divided into two-phases: offline (performed by the service provider), and online (which involves both parties, service provider and user).

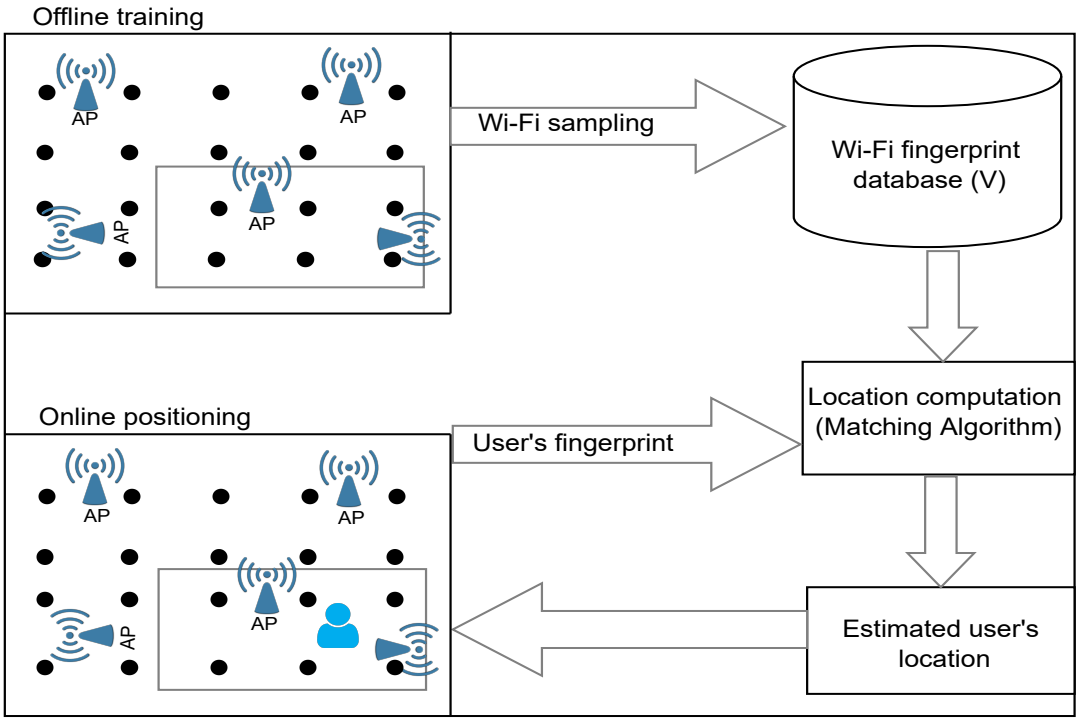


Figure 5.1: System Architecture Illustrating the Pre-Processing Phase (Offline Phase), in Which the Wi-Fi Signals (RSS) Are Recorded for Each Reference Point (The Black Dots) by the Service Provider, and the Online Positioning Phase, in Which the User Is Localised Based on the Submitted Wi-Fi Signals Using the Matching Algorithm. The AP Represent the Wi-Fi Access Points, Also Known as the RSS Values.

In the offline phase, the SP records the Wi-Fi fingerprinting known as received signal strength in an area assumed to be a rectangular grid of points of interest (x_i, y_i). Each location point (x_i, y_i) where $i = 1 \cdots l$ falls within a range of an Access Points (AP), and the pair (x, y) is the location coordinate of the fingerprint. The RSS $\hat{n}$-dimensional vector $V_i = \{r_1, r_2, \cdots, r_{\hat{n}}\}$ where $\hat{n}$ is the total number of access points within the range of a location (x_i, y_i) , and $\hat{n} \leq n$. In reality, $\hat{n}$ will always be less than n, because no location (x_i, y_i) can be within the range of all access points n. A database (radio map) is created as a reference point to localize a user during the online phase. An example of the database is shown in Table 5.2. In practice, the RSS signal quality measured in decibels is considered exceptional when it's below -40, very good between -40 to -55 and very low beyond -80. We assign -100 to indicate an access point is unavailable.
In the online phase, the user using a positioning algorithm, gathers RSS values $V_{u_i} = \{r_1, r_2, \cdots, r_{\hat{n}}\}$ at a location (x_i, y_i) . These RSS values will be compared with RSS tuples $V_i = \langle \{r_{ij}\}_{j=1}^{\hat{n}} \rangle_{i=1}^{l}$ for l locations and n access points in the database of the SP.

Table 5.2: Wi-Fi Fingerprint Reference Database

l_i	ap_1	ap_2	ap_3	ap_4	$\cdots$	ap_n
1	$r_{1,1}$	$r_{1,2}$	-100	-100	$\cdots$	$r_{1,n}$
2	-100	$r_{2,2}$	$r_{2,3}$	$r_{2,4}$	$\cdots$	-100
3	-100	-100	$r_{3,3}$	$r_{3,4}$	$\cdots$	$r_{3,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
l	$r_{l,1}$	$r_{l,2}$	$r_{l,3}$	-100	$\cdots$	-100

The Euclidean distance θ_i, which is a measure between V and V_u is then computed to localize the position of the user. There are various distance algorithms (Manhattan distance, Minkowski distance, *etc.*) that can be used, but in this work, we use the Euclidean distance :

$$\theta_i = \|V_i - V_{u_i}\| = \sqrt{\sum_{j=1}^{n}(V_i - V_{u_i})^2} \tag{5.1}$$

Due to the difficulty in making computations involving the square root of encrypted values, the squared Euclidean distance between V and V_u is rather used in our algorithm as it achieves the same result:

$$\begin{aligned} \theta_i = \|V_i - V_{u_i}\|^2 &= \sum_{j=1}^{n}(r_{ij} - r_{u_i})^2 \\ &= \sum_{j=1}^{n} r_{ij}^2 + \sum_{j=1}^{n} r_{u_i}^2 + \sum_{j=1}^{n} - 2r_{ij} \cdot r_{u_i} \end{aligned} \tag{5.2}$$

The k nearest neighbours $\{\theta_i\}_{i=1}^{k}$ are then selected and the average computed to localize the position of the user.

5.2.1 The Two-Party Protocol

The description of the Wi-Fi-indoor location algorithm didn't factor any privacy concern and the computational limitations at the user's side.

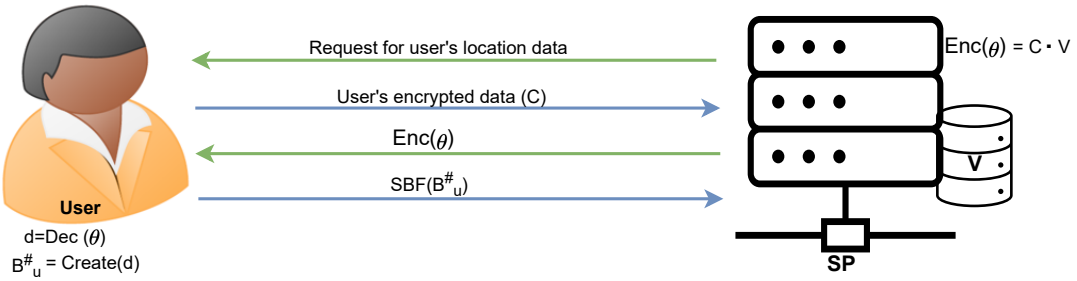


Figure 5.2: System Design for Two-Party Computation With Private Inputs. The Light Green Represents Data From the Service Provider (SP), While Data From the User Is in Light Blue.

In this section, we present a novel algorithm where all these concerns are taken into consideration. In addition, unlike other location algorithms, where applications are based on the user's exact location, here the SP tries to learn the location of the user without violating their privacy: in particular, the protocol will never reveal the user's exact location to the service provider, but only whether the user is within any area of interest. Areas of interest are pre-defined by the SP (before execution of the protocol). If the user is outside such an area, the provider learns nothing.
The two-party computation involves the user and the SP, and the communication happens over an authenticated channel. That is, both sides are sure the messages they are receiving are coming from the right source. With this protocol, the SP is interested in learning the location of the user but doesn't want the user to have access to the database used in computing the location, whereas the user doesn't want the SP to learn the exact location but only presence in a predefined area of interest. This kind of problem is called secure two-party computation where both sides cannot learn anything about their respective inputs, beyond the explicit output.

Protocol 1 Privacy Preserving Protocol for Indoor Wi-Fi

User		**Server**
$\mathsf{Enc}(B^{\#})$		$\mathsf{Enc}(B^{\#}),\ V$
$V_u \leftarrow\$ \{0,1\}$		
$C \leftarrow\$ \mathsf{Enc}(\mathsf{pk}_u, V_u)$		$\mathsf{Enc}(B^{\#}) \leftarrow\$ \mathsf{Enc}(\mathsf{pk}, B^{\#})$
	$C = \{c_j\}_{j=1}^{\hat{n}} \longrightarrow$	
		$\mathsf{Enc}(\theta_i) \leftarrow\$ \mathsf{Enc}(V_u) \odot V$
	$\{\mathsf{Enc}(\theta_i)\}_{i=1}^{k} \longleftarrow$	
$\theta_i \leftarrow\$ \mathsf{Dec}(\mathsf{sk}_u, \mathsf{Enc}(\theta_i))$		
$d = (x,y) = loc\ \{\theta_i\}_{i=1}^{k}$		
$B_u = create(d)$		
$B_u^{\#} = \mathsf{Enc}(B^{\#}) \odot B_u$		
	$B_u^{\#} \longrightarrow$	
		$\mathsf{Dec}(\mathsf{sk}, B_u^{\#})$
		$i \leftarrow\$ \{\triangle_i\}_{i=1}^{s}$
		$0 \leftarrow\$ Otherwise$
finalize		

The proposed Protocol 1 addresses the concerns of both parties by revealing only the area of the user's location but nothing else to the SP. The nature of the SBF filter, a variant of Bloom filter [Blo70] only makes it possible for binary input. That is, given a vector of n -bits initially set to zeroes, k-independent hash functions are used to map locations (elements) unto the filter by setting their vector corresponding positions to 1. The user then shuffles these positions to prevent the SP from detecting the exact location of the user. On the flip side, the SP's database is also protected from malicious users.

We now give a detailed description of the protocol. The protocol assumes a planar rectangular grid covering finite points of interest (PoI) with a space of $1\ m$ between two grid points. Each point of interest is within the range of $\hat{n}$ access points of a total of n access points in the building or system.

In the offline phase, SP records the Wi-Fi fingerprints multiple times at each location for each access point and the average is computed. Thus r_i will be the average of r_{ij}. Thus if $j = 3$, then r_i will be the average of measuring three times for each access point at position (x_i, y_i). Like Table 5.2, the database is constructed for all points of interest.

Algorithm 2 Spatial Bloom Filter Algorithm

Input : $\{\triangle_i\}_{i=1}^{s},\ H \leftarrow\$ \{h_j\}_{j=1}^{q}$
Output : $B^{\#}$

1 : **for** $i \leftarrow\$ 1 \cdots s$ **do**
2 : **for** $j \leftarrow\$ 1 \cdots q$ **do**
3 : **foreach** $t \in \triangle_i$ **do**
4 : $B^{\#}[h(t)] \leftarrow\$ i$
5 : **return** $B^{\#}$

The SP constructs a spatial Bloom filter (SBF) [PCM14, CPM15, CPM18], a variant of the Bloom filter [Blo70] and a data structure that allows spatial representation. Like the original Bloom filter, values are inserted using a one-way function that supports probabilistic membership queries, that is, whether a member belongs to the set or not. The SP constructs an SBF over only areas of interest, then encrypts the filter $\mathsf{Enc}(B^{\#})$ and sends it to the user.
The online phase is divided into four sub-phases: measuring, computation, private localization, and sever localization. In the measuring phase, a user u, using a positioning algorithm, obtains a fingerprint V_u and uses Pailler's cryptosystem to generate a pair of keys $< pk_u, sk_u >$. The key generation can be executed once and offline, and the secret key can be kept until it expires (or is compromised).
When the user receives a request from the SP, the user queries the local dataset V_u and uses the pk_u to encrypt each r_{ij} and sends it to the server as outlined in algorithm 3.

$$\begin{aligned} C &= \mathsf{Enc}(\mathsf{pk}_u, V_{u_j}) \\ &= \{c_1, c_2, \cdots, c_{\hat{n}}\} \end{aligned} \tag{5.3}$$

$$c_j = \{\mathsf{Enc}(pk_u, -2r_j),\ \mathsf{Enc}(pk_u, r_j^2)\} \tag{5.4}$$

Algorithm 3 Private Localization of User

Input : $V_u = \{r_j\}_j^{\hat{n}} \leftarrow\$ \{0,1\}^n$

Output : $t_u = (x, y)$

User Outsourced Location

1 : **for** $j \leftarrow\$ 1 \cdots \hat{n}$ **do**

2 : $\quad \mathsf{Enc}(V_{u_j}) \leftarrow\$ \mathsf{Enc}(\mathsf{pk}, r_j)$

3 : $post\ \mathsf{Enc}(V_{u_j})$

User Private Location

4 : **get** $\{\mathsf{Enc}(\theta_i)\}_{i=1}^{k}\ from$ **S**

5 : **for** $i \leftarrow\$ 1 \cdots k$ **do**

6 : $\quad \theta_i \leftarrow\$ \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\theta_i))$

7 : $\quad t_u = (x, y) \leftarrow\$ mean\{\theta_i\}$

8 : $B_u = create(t_u)$

9 : $B_u^{\#} = \mathsf{Enc}(B^{\#}) \odot B_u$

10 : **post** $B_u^{\#}$ *to* **Server**

The computation phase: after the localization server had received the encrypted fingerprint $\{c_i\}_{i=1}^{\hat{n}}$, it uses the Pailler's homomorphic additive properties to compute the Euclidean distance θ_i using a set of V_i from the reference database to obtain the encrypted k-nearest neighbours. As a further protection to the database against a malicious user, the server permutes the distance ciphertext and add a generated random number to it. The computation under encryption is similar to Equation 5.2. Basically, the server encrypts the squares of the selected training fingerprint r_{ij}^2 to output:

$$\begin{aligned} \mathsf{Enc}(\theta_i) = \prod_{j=1}^{N} \mathsf{Enc}(r_{ij}^2) \cdot \mathsf{Enc}(r_{u_j}^2) \cdot \mathsf{Enc}(-2r_{ij} \cdot r_{u_j}) \\ where \quad \mathsf{Enc}(-2r_{ij} \cdot r_{u_j}) = \mathsf{Enc}(-2r_j)^{r_{ij}} \end{aligned} \tag{5.5}$$

Private location retrieval phase: the user receives the encrypted k-nearest neighbours $\mathsf{Enc}(\theta_i)$ line 4 of algorithm 3, decrypts it with the private key sk_u to obtain the K nearest neighbours θ_i same as Equation 5.1. The user then estimates the private location d by computing the geometric center of the k-nearest

neighbours.

Sever localization phase: in the localization, the user first uses algorithm 2 to construct an SBF over the exact location d.The user who has access to SP's encrypted SBF $\mathsf{Enc}(B^{\#})$ over the area of interest (AoI) will perform entry-wise homomorphic product of the encrypted SBF $\mathsf{Enc}(B^{\#})$ and B_u to obtain $B_u^{\#}$ as illustrated in algorithm 3. The user then randomizes the order of values in the filter by shuffling $\mathsf{Enc}(B^{\#})$, the output of the encrypted filter and sends it to the server.

When the server receives the encrypted filter $\mathsf{Enc}(B^{\#})$, it uses the private key $\boldsymbol{sk}_s$ to decrypt the filter and performs location verification using algorithm 5. Should any of the hash function $h \in H$ return 0, then certainly the user is not in an area of interest.

Algorithm 4 Localizing User's Location by Server

Input : $\{\mathsf{Enc}(V_u)\}_{j=1}^{\hat{n}},\ \{V\}_{i=1}^{n}$

Output : $\triangle_i$

Computing KNN in Encrypted Format

1 : **for** $j \leftarrow\$\ 1 \cdots n$ **do**
2 : **for** $j \leftarrow\$\ 1 \cdots \hat{n}$ **do**
3 : $\theta[i] \leftarrow\$\ \mathsf{Enc}(V_{u_j}) \odot V_i$
4 : $post \quad \{\theta_i\}_{i=1}^{k}$

Importing User Location

5 : **get** $\mathsf{Enc}(B_u^{\#})\ from\ User$
6 : **for** $i \leftarrow\$\ 1 \cdots S$ **do**
7 : **if** $\triangle_i = \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(B_u^{\#}))$
8 : **return** $\triangle_i$
9 : **else** :
10 : **return** 0

But if the output bits are greater than 0 for all hash functions, and the smallest output value is i, then the user's position is $d \in \triangle_i$ minus the false probability of the filter.

Algorithm 5 Verification of Spatial Bloom Filter Algorithm

Input : $B_u^{\#},\ H \leftarrow\$ \{h_j\}_{j=1}^{q}, \mathbf{B} = \{\triangle_i\}_{i=1}^{s}$

Output : $\triangle_i$

1 : **for** $i \leftarrow\$ 1 \cdots s$ **do**
2 : **for** $j \leftarrow\$ 1 \cdots q$ **do**
3 : **if** $B^{\#}[h(t_u)] = 0$:
4 : **return** 0
5 : **else** $B^{\#}[h(t_u)] \leftarrow\$ i$:
6 : **return** $\triangle_i$

5.3 Security

In this section, we present security definitions for the two-party computation protocol, based on the *honest-but-curious* model [CDN15]. Thus, the parties follow the instructions of the protocol but may try to learn more information than allowed by the protocol.
The two-party protocol is secure if, at the end of execution, the exact location of the user is privacy preserved and the server learns only the predefined area of interest of the user's location but not the exact location. If the user is outside the predefined area of interest, then the server learns nothing about the user's location. The protocol preserves the service provider's privacy if, at the end of the protocol execution, the user is unable to obtain the list of the reference Wi-Fi fingerprint (radio map), and the privacy of the predefined areas encoded in the filter is preserved.

5.3.1 Security Discussion

To begin with, the user's fingerprint sent to the server is encrypted using the semantically secure Paillier's cryptosystem. Hence the server cannot determine the closest neighbours to the location of the user without the user's private key sk_u. In addition, the exact location of the user is protected from the server. This is achieved by using algorithm 3 line 9, by encoding the exact location of the user using (spatial) Bloom filter to produce an encrypted filter $B_u^{\#}$. Before the user sends $B_u^{\#}$ to the server, it permutes the filter by shuffling it to change the order of the values corresponding to the user's location. After decryption by the

server, it will be impossible for the server to reconstruct the user's filter based on the order. Also, the security properties of Paillier's encryption algorithm (or any other additively homomorphic encryption scheme) ensure that the server learns only limited values of $B^{\#}$ corresponding to the non-zero values of B_u [KS14], making it impossible to identify the user's exact location. The server will learn only the area of interest i if the number of non-zeroes corresponds to b and all the values take i as the area of interest. If the number of non-zero values corresponds to b, but some of the values are $> i$, due to the collision of the filter, then the server learns only the area of interest $\triangle_i$ and some patterns of values but not the exact location. However, if the number of non-zero values is $< b$, after the server decrypts $B_u^{\#}$ using algorithm 4 line 7, then certainly the server learns nothing about the user's location, making it privacy preserved.
We argue the protocol also preserves the privacy of the SP data. In that, anytime the user queries for the K nearest neighbours, the server blinds the result with a random number that introduces enough noise to prevent the user from knowing the real distances. Also, the permutation prevents the user from knowing the indexes of the closest neighbours. Though some information may be leaked after computing the centroid, this is insignificant. To a larger extent, the database is protected so far as both parties follow the instructions of the protocol. The privacy of the predefined areas is preserved thanks to the homomorphic encryption of the filter. From algorithm 3 line 9, the user can perform multiplication without knowing the cleartext of the predefined areas of interest inserted in the filter by the server.
We do acknowledge that a malicious user may attempt to build its own database by recording enough RSS values over the same field. However, as we can assume the area is controlled, this will be practically unfeasible. Moreover, the malicious user needs almost the same amount of RSS values to make accurate predictions.

5.4 Conclusion

In this chapter, we developed a robust privacy-preserving cryptographic protocol for indoor Wi-Fi localization that effectively addresses the privacy concerns of both users and service providers. The protocol ensures that the service provider remains unaware of the user's precise location outside pre-defined sensitive areas while simultaneously preserving the confidentiality of the service provider's database. As a result, neither party can access information about the

other's input beyond what the protocol allows. Thus, the chapter demonstrates a successful well-balanced solution for preserving location privacy in indoor Wi-Fi localization without compromising functionality. The main conclusion to the chapter can be found in 7.1.2.

Chapter 6

A Cryptographic Protocol for Efficient Mutual Location Privacy Through Outsourcing in Indoor Wi-Fi Localization

In this chapter, we present an advanced cryptographic protocol for efficient mutual location privacy through outsourcing, building on and refining the previously presented two-party privacy-preserving protocol for indoor Wi-Fi localization. The work has been accepted for publication in IEEE Transactions on Information Forensics and Security and is available under the "Early Access" area on IEEE Xplore [EP24].

6.1 Introduction

This chapter addresses the limitations and improves upon the two-party privacy-preserving protocol introduced in the preceding chapter for indoor Wi-Fi localisation. As digital services and applications are individualised and tailored to each user based on the user's context [GM02], it is crucial that the user's privacy is protected without compromising security. The LBS and other location-aware applications are good examples of such tailored services. Most LBS providers rely on user devices, such as smartphones equipped with a satellite-based Global Navigation Satellites System (GNSS) receiver, to collect this location information. Examples of GNSS are Europe's Galileo, USA's Global Positioning System (GPS), Russia's Global Navigation Satellite System (GLONASS) and China's BeiDou Navigation Satellite System (BDS). Unfortu-

nately, as stated earlier, the satellite-based navigation systems' positioning signals are too weak in an indoor environment and can only be relied upon when used outdoors. Because of this challenge, the Wi-Fi access point would be adopted due to the underlining advantages, as stated in the previous chapter. However, the collection of location information raises a number of privacy concerns that the Wi-Fi localisation protocol previously introduced in Chapter 5 attempted to address. Though this protocol is robust in protecting user privacy, the protocol only assumes a *semi-honest* adversary security model. This type of adversary follows the instructions of the protocol execution but may attempt to gather more information about the protocol without playing any active role. As a result, the information is gathered mainly through observation, such as through analysing the protocol's metadata. While this assumption may be useful in certain situations, it may not accurately reflect the potential threats posed by active corrupt adversaries (malicious adversaries). Specifically, when one considers environments that are more complex and unpredictable, such as the IoT, it is more realistic in today's cybersecurity landscape that threats from active adversaries must always be incorporated in the design of such a privacy-preserving localization protocol. However, this is the case even if one does not consider the IoT a complex or unpredictable environment.

It is crucial to emphasise that, given the variety of real-world applications and the various threat scenarios they entail, these two protocols — the semi-honest and active adversaries settings offer different functionalities. For example, the semi-honest model is suitable in an environment where entities are generally perceived to be trustworthy. Thus, the primary concern is the exposure of sensitive information to passive adversaries, typically necessitating a balance between usability and security. On the other hand, models considering active adversaries are indispensable in hostile or complex environments where deliberate or active attacks are more likely. In such scenarios, it is imperative to deploy robust security measures to protect against active adversaries.

To protect against active(malicious) adversaries, our protocol introduces a third party in the design phase and, in addition, utilizes a combination of cryptographic primitives. These include techniques such as homomorphic encryption, which allows for the computation of encrypted data without decrypting it, and secure multi-party computation, which enables multiple parties to jointly compute a function without disclosing their inputs.

For a quick recap, generally, the process of localizing a user in an indoor space using Wi-Fi comprises two phases: offline and online. In the offline phase, a

service provider (SP) builds a fingerprint reference database called *radio-map*, by measuring the Wi-Fi *received signal strength* (RSS) in a number of reference locations (points for which exact location coordinates are known). The reference points are normally chosen following a geometrical grid structure. For each reference point, the set of different RSS values for the available access points becomes the fingerprint of that point. During the online phase, a user generates the fingerprint of their location by measuring the RSS signals from all the access points within the user's range. The fingerprint is then sent to the service provider to compute the user's location. The location is calculated by comparing the received fingerprint to the reference database built in the offline phase, using any distance algorithm. Distance metrics include Euclidean distance, support vector machine, likelihood estimator, Sorensen distance and the Manhattan distance [TMT+15].

The user-centred personalization of digital services introduces privacy challenges, as personalization depends on the user's information being disclosed. Preserving privacy in location-based services is particularly important due to the high degree of uniqueness of human location traces, which prevents anonymity and makes user re-identification trivial [dMHVB13b]. Preserving location privacy in indoor LBS is particularly challenging: unlike GNSS-based applications, where GNSS receivers can compute the location without any interaction with another party, Wi-Fi fingerprinting typically requires the service provider's assistance in determining the user's location. This, in turn, implies disclosure of the user location information to the server. The simplest solution to this problem would be to grant the user full offline access to the reference database to locally compute the location. However, this trivial solution not only burdens the already constrained computational resources of the user device (smartphone or IoT device) but also requires disclosure of information in the opposite direction (from the provider to the user), potentially creating an opposite privacy issue if the service provider's database contains sensitive locations. In addition, disclosure of the radio map could leak sensitive information about critical infrastructures, such as airports or hospitals, which often are primary users of Wi-Fi indoor localization. It could also reveal the provider's areas of interest, that is, selected sensitive areas in the indoor space. For these reasons, in this work, we focus on *mutual location privacy*, which aims to ensure the privacy of the sensitive information held by both the user (their location) and the service provider (the radio map and areas of interest). This is a specific realization of the general concept of secure multi-party computation.

Secure multi-party computation (MPC) is a cryptographic technique that allows parties to jointly compute a function without revealing their private inputs in order to preserve their privacy. In indoor Wi-Fi localization protocols, MPC can be used to enable location-based services while preserving the privacy of the users. However, the computation required for MPC can be resource-intensive, which may limit its scalability and practicality, especially for clients with limited computational power. In addressing this bottleneck, outsourcing client computation to a cloud server can offer several benefits. One of the main advantages is that it allows clients to participate in the computation without being an impediment in the process. Another benefit is that cloud servers can provide high computational power and storage capacity that can improve the accuracy and efficiency of indoor localization protocols making cloud server outsourcing an attractive option for clients. Recent studies have demonstrated the effectiveness of outsourcing computation to the cloud in secure multiparty computation [JLL+19, OPS22, ZHS+22]. These studies highlight the potential of cloud outsourcing in enabling scalable and efficient indoor localization with enhanced privacy and security using secure multiparty computation.

An area yet to be fully addressed in the literature is how a service provider can securely check if a user is present in some predefined areas of interest without learning their location outside those areas. For instance, an employer may wish to have real-time access to employees' location in predefined areas of interest (e.g. to verify attendance), while employees have an interest in preserving their privacy elsewhere in the building or outside the company premises [Gro04]. Another scenario is related to healthcare: a hospital may want to collect information on the average time a patient spends in a specific ward receiving a particular service. The same might apply to health insurance companies, who might want to compare hospitals to inform their reimbursement portfolio, i.e., reimbursing based on value [CFFM15]. In order not to infringe on their privacy, a patient should only be monitored when they enter certain parts of the building (e.g. a scan room), and the provider should learn nothing about their location elsewhere; at the same time, the map and indoor locations of these parts of the building may need to be kept concealed for security.

In certain instances, it may be necessary to track the movement of individuals within a building, such as during an emergency evacuation. However, the privacy of the individuals must be protected, and their movement outside of the designated areas of interest should not be monitored. WiFi localization can be used to ensure that the data collected is only pertinent to the area of interest

and that the movements of users outside of the area of interest remain anonymous. In addition, in the wake of the COVID-19 pandemic, contact tracing has become a vital tool for controlling a virus's spread. One use case in our protocol settings is the identification of individuals who have been in close proximity to an infected person within a specific area; therefore, the locations of such individuals are not to be monitored in the building when they are outside of the infected person's area. This can help protect the privacy of individuals while enabling effective contact tracing.

6.1.1 Related Work

In the following, we give an overview of relevant works in the literature related to preserving the privacy of location information. Most of the works fall under two broad categories based on their approach: homomorphic encryption and combining two secure computations.

Our work applies to the location privacy setting the concept of outsourced secure multi-party computation. The validity and usefulness of this concept and setup have been demonstrated in other settings in several works, including Araki et al. [AFL+16]. In their three-party computation protocol, designed for network authentication (Kerberos), security is achieved against malicious clients, and privacy can be guaranteed in the presence of two semi-honest adversaries (servers).

A construction based on homomorphic encryption was proposed by Li et al. [LSZ+14], who designed a protocol for privacy-preserving indoor localization that utilizes Paillier encryption to protect the parties involved. Their security model only considers an *honest-but-curious* client in protecting the service provider's database. In fact, results [YJ18] show that an attacker could carefully craft a packet that can leak the service provider's entire database. Zhang et al. [ZCZL16] adopt a more advanced approach to using homomorphic encryption (again based on the Paillier cryptosystem), which involves a support vector machine for regression (SVR) to protect the untrusted parties involved in the protocol. However, even this protocol is not secure against a malicious client: a specifically crafted query by a client can easily expose the service provider's decision function used in training the data set. The work [EP19] presented an indoor localization protocol with privacy. Unlike most other works that focus on how the client can access their location privately, this work permits the service provider to determine privately if a user is within predefined areas of interest. Therefore, the client can privately learn their exact location while allowing the

service provider to query if the user is within an area of interest. However, this work is also limited by considering only honest-but-curious users.

The second type of protocol combines two secure computations to preserve the privacy of the location information. An early example of such protocols is the work by Shu et al. [SCYW14] that combines Paillier encryption and oblivious transfer to design a privacy-preserving localization protocol. The protocol is designed to protect users' location and that of *anchor helpers* (similar to reference points), who assist in computing a user location where the main user device is not equipped with location sensors. For example, a laptop (the main client) may rely on a nearby smartphone (the anchor helper) to determine their position. Unfortunately, in [SCYW14], the resource constraint mobile client does all the heavy computations, limiting the scheme's real-world applicability. Yang et al. [YJL21] also present a privacy-preserving location framework for proximity detection in their work. By using an effective homomorphic encryption mechanism, the algorithm enables users (queriers) to discreetly query an LBS agent for a range of arbitrary polygons (search regions). Five entities are involved in the proposed protocol (PRQ), including the querier and a trusted party. All other entities in the protocol's security model are semi-honest except for the trusted party.

Recent work by Järvinen et al. [JLL+19] presents a privacy preservation indoor localization protocol (PILOT). The construction, which is implemented using the ABY framework [DSZ15], combines secure computations (like secret sharing and Yao's garbled circuit) to preserve the client's privacy and the database hosting server. The protocol outsources the computations to two non-colluding semi-honest parties different from the server hosting the database. Oleynikov et al. [OPS22] recently proposed an LBS privacy-preserving proximity testing protocol called POLAR. The algorithm, in particular, enables two clients (client-to-client) to outsource most of their heavy computing to two servers by leveraging Secure Multi-Party Computation (SMPC) to improve the efficiency of the proposed protocol. In addition, the protocol is secure against a static adversary with at least one uncorrupted server.

The work of [YVSG22] employs 1-out-of-n oblivious transfer for ensuring privacy in location-based navigation. In addition, their work utilises private information retrieval techniques due to the server's caching of the user's keywords, which reduces the communication overhead.

Some recent works try to augment the capability of Wi-Fi to localise users by employing additional information sources. For instance, Zhang et al.[ZHC+22]

recently presented a private indoor localization strategy that uses both Wi-Fi and Bluetooth signals. To protect the fingerprint, their approach applies a differential privacy mechanism based on semi-supervised machine learning. The authors of [RHWL22] propose an indoor localization technique that employs a commercial Wi-Fi chipset to extract features such as motion and breathing patterns. Their technique achieved an accuracy of 98.5% for a single user: this result demonstrates the urgent need for privacy protocols that guarantee the protection of the user's location.

Most privacy-preserving protocols in the literature focus on preventing leakage of the service provider's database and client location while still allowing the client to seek the service provider's assistance for the localization service. In this chapter, we maintain the privacy guarantees of the user while at the same time increasing the capabilities and privacy of the service provider (SP). In particular, the proposed protocol allows the service provider to securely verify if a user is present in certain limited and predefined areas of interest while maintaining the user's privacy elsewhere. Like others in the literature, the protocol also allows users to securely and effectively seek the service provider's assistance for private localization.

Our protocol involves non-colluding servers, as explained in the design of the security model (Section 6.3). The assumption of non-collusion has been widely used in the literature, including [RVBM09, SS13, KMRS14, JLL+19]. The assumption of a non-colluding service provider and the cloud server is based on the idea that there are scenarios in which both parties have incentives to keep their clients' data secure and private, as discussed in the above works. Among these is that both the SP and the CS are often contractually required to provide a secure and private service. Any breach of this contract may result in losing the client's trust, in turn resulting in a loss of business and reputation. Because service providers generally build their business model on the trust of their clients, any breach could result in serious legal and financial consequences. While collusion is always possible, the risks and incentives associated with maintaining client data privacy and security strongly motivate both parties to follow the protocol and not collude. Furthermore, the cloud server and service provider may be required by law or regulation to protect the privacy and security of their client's data. In practice, there are numerous scenarios in which a service provider and a cloud server are not colluding. One example is when a business outsources its data processing requirements to a third-party cloud service provider. The cloud service provider may perform computations on the

company's behalf, but the company retains ownership of the data and computations.

6.1.2 Contribution

This chapter presents an efficient and secure indoor location privacy protocol that allows the service provider to query if a user is within an area of interest (out of a predefined set of areas) while maintaining the exact location of the user's privacy. As such, the provider does not learn the location of the user but only the identifier of the area of interest if the user is within one and nothing otherwise. The scheme prevents providers from defining too many or too broad areas, while the provider's database is also protected from malicious users.
The proposed protocol supersedes previous works in the indoor Wi-Fi location privacy domain, including [EP19], which was presented in the previous chapter, where the protocol offers protection to both the user's fingerprint and the service provider's database yet delivers the service based on the users' physical location. Unlike [EP19], where the security model was limited by the assumption of honest-but-curious users, this protocol's security is assured even against a malicious user. Our work combines state-of-the-art techniques in secure multi-party computation to preserve the privacy of the SP database against malicious users. The model builds upon the spatial Bloom filter data structure [PCM14, CPM15, CPM18] and substantially improves the overhead at the client-side with respect to comparable protocols in the literature. Specifically, the main contributions of this chapter are summarized below:

- We design a hybrid technique by combining Damgård, Geisler and Krøigaard (DGK) [DGK07, DGK09] encryption and an optimized garbled circuit to preserve both parties' privacy. DGK encryption protects the user's query data and significantly reduces the message size space compared to other well-known additive homomorphic encryption schemes like Paillier (used, for instance, in [AFL+16]). The garbled circuit [Yao82, Yao86], on the other hand, preserves the privacy of the database against malicious users.
- Our protocol provides protection against malicious users with the addition of a single server (CS). This method is a significant improvement over comparable protocols, such as [JLL+19], which requires two parties to achieve comparable security levels.
- Our protocol significantly improves the computation overhead for encrypting the fingerprint compared to solutions in the literature (as dis-

cussed in Section 6.4) and offloads almost all the computation overhead at the retrieval phase to the cloud server. In addition, the communication overhead is reduced by over 99.99% at the client-side compared to the non-outsourced protocols. We achieved this by delegating most of the heavy computation of the garbled circuit at the user-side to the cloud server, thus making the protocol suitable for practical applications when compared to computationally intensive client/server proposals in [LSZ+14, EP19, YJ18].

- We provide a security and complexity analysis of our protocol and quantify the computation and bandwidth gains compared to other similar localization protocols available in the literature (Sections 6.3 and 6.4).

Table 6.1: Notation: Acronyms and Symbols.

Symbol	Meaning
RSS	Received signal strength
n	The number of access points (ap)
m	The number of reference points
$\boldsymbol{r_{i,j}}$	RSS value at position i for ap j
$\boldsymbol{V_i}$	RSS values for all n ap's at i
$\boldsymbol{V_u}$	RSS values of user (u)
$\boldsymbol{c}$	Encrypted RSS values of user
SBF	The spatial Bloom filter
$\triangle$	An area of interest (AoI)
$\boldsymbol{\mathcal{T}_2}$	List of reference points
$\boldsymbol{\xi}$	Set of all the areas of interest
$\boldsymbol{B^{\#}}$	SBF over the set of areas of interest
$\mathsf{Enc}(\boldsymbol{B^{\#}})$	Encryption of $B^{\#}$ using pk_s
$\boldsymbol{pk_u}$	Public key of user U
$\boldsymbol{sk_u}$	Secret key of user U
$\boldsymbol{pk_s}$	Public key of service provider SP
$\boldsymbol{sk_s}$	Secret key of service provider SP
T	The bit-length of RSA moduli
$\mathsf{Enc}(\boldsymbol{\delta})$	Unmasked encrypted euclidean distance
$\mathsf{Enc}[\boldsymbol{\delta}]$	Masked encrypted euclidean distance
$\mathbb{D}$	Referenced database of service provider
$\boldsymbol{\pi}$	Garbled indexes of smallest KNN distances
$\boldsymbol{\gamma}$	Random noise/value
$\boldsymbol{d}$	User exact location coordinate
$\boldsymbol{B_u}$	Bloom filter build over $\triangle$
$\boldsymbol{B_u^{\#}}$	Output of $\mathsf{Enc}(B^{\#})$ and B_u

6.2 Private Indoor Wi-Fi Localization

In this section, we propose a novel private protocol for indoor Wi-Fi-based localization. In the following, we assume the reader has a good knowledge of the main primitives used in the construction: homomorphic encryption and, in particular, the cryptosystem of Damgård, Geisler and Krøigaard (DGK); Oblivious Transfer (OT); Garbled Circuits (GC); Bloom Filters (BF) and Spatial Bloom Filters (SBF). For reference, we provide an introduction to these primitives in Chapter 2 and Table 6.1, which outlines the notation and symbols used in this chapter.

The protocol involves three parties: a user, the provider of the location-based service, and a third party to which complex computation on the user side is outsourced. The protocol security is discussed in Section 6.3, while its efficiency against existing solutions is evaluated in Section 6.4.

As described in the introduction, the process of Wi-Fi fingerprinting is generally divided into two phases: the offline set-up phase (usually run by the service provider SP) and the online phase.

In the set-up phase, the SP creates a reference database $\mathbb{D}$, by generating a fingerprint (V) for each reference point. The fingerprint represents the received signal strength (RSS) for each access point within the reference point range. For a given reference point (x_i, y_i), an n-dimensional vector $V_i = \{r_1, r_2, \cdots, r_n\}$ is generated, where n is the total number of access points. The total number of reference points is m.

The fingerprints for all reference points together constitute the reference database (radio map) which is later used to localize a client during the online phase. Table 6.2 depicts an example of the reference database. Symbol $\perp$ indicates that the reference point is outside of the range of the access point (that is, no signal is received for that access point).

During the online phase, a user u wishing to obtain their location generates a fingerprint by measuring the RSS values $V_u = \{r_1, r_2, \cdots, r_n\}$ at their location (x_u, y_u). The user's fingerprint is then compared to the reference database $\mathbb{D}$ to retrieve the position of the user with the help of any distance computing algorithm.

The Euclidean distance, represented in the equation 6.1, is one of the most used distance algorithms. It measures the difference in distance between V and V_u.

$$\delta = \|V - V_u\| = \sqrt{\sum_{i=1}^{n} (V_i - V_{u_i})^2} \tag{6.1}$$

Table 6.2: Wi-Fi Fingerprint Reference Database ($\mathbb{D}$): Each Column Relates to an Access Point ap_i, While Each Row Represents the Fingerprint V_i for Reference Point Coordinates (x_i, y_i). $\perp$ Indicates No Signal From the ap at That Location. $\mathbb{D}$ Is Also Called a Radiomap.

	ap_1	ap_2	ap_3	ap_4	$\cdots$	ap_n
V_1	$r_{1,1}$	$r_{1,2}$	$\perp$	$\perp$	$\cdots$	$r_{1,n}$
V_2	$\perp$	$r_{2,2}$	$r_{2,3}$	$r_{2,4}$	$\cdots$	$\perp$
V_3	$\perp$	$\perp$	$r_{3,3}$	$r_{3,4}$	$\cdots$	$r_{3,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
V_m	$r_{m,1}$	$r_{m,2}$	$r_{m,3}$	$\perp$	$\cdots$	$\perp$

As stated earlier, a computation involving the square roots cannot be directly rendered through homomorphic computation; hence the squared Euclidean distance (equation 6.2) which achieves the same result, is used here instead.

$$\begin{aligned}\delta = \|V - V_u\|^2 &= \sum_{i=1}^{n}(r_i - r_{u_i})^2 \\ &= \sum_{i=1}^{n} r_i^2 + \sum_{i=1}^{n} r_{u_i}^2 + \sum_{i=1}^{n} - 2r_i \cdot r_{u_i}\end{aligned} \tag{6.2}$$

From equation 6.2, the smallest k nearest distances $\{\delta_j\}_{j=1}^{k}$ are selected and the average (k-means) is computed to localize the user.

6.2.1 Protocol Participants

Three parties participate in the proposed protocol, as described below.

User: a mobile device accessing a service provider to obtain his/her exact location by inputting the related RSS fingerprint V_u.

Service Provider (SP): the location service provider who wishes to know if a user is within an area of interest $\triangle$ (defined over a geographical grid) without infringing the user's privacy by learning the location outside of the area of interest. Even when the user is located within the area of interest, the server only learns this fact (presence in the area), and not the user's exact location. The server can define a set of areas of interest $\xi = \{\triangle_1, \ldots, \triangle_l\}$.

Cloud Server (CS): a third party who aids the user by performing computationally expensive operations without making any input to the evaluation function.

6.2.2 Protocol Overview

Here, we provide a high-level overview of the protocol. Later in the text, the protocol is detailed further by identifying the operations performed by each participant.

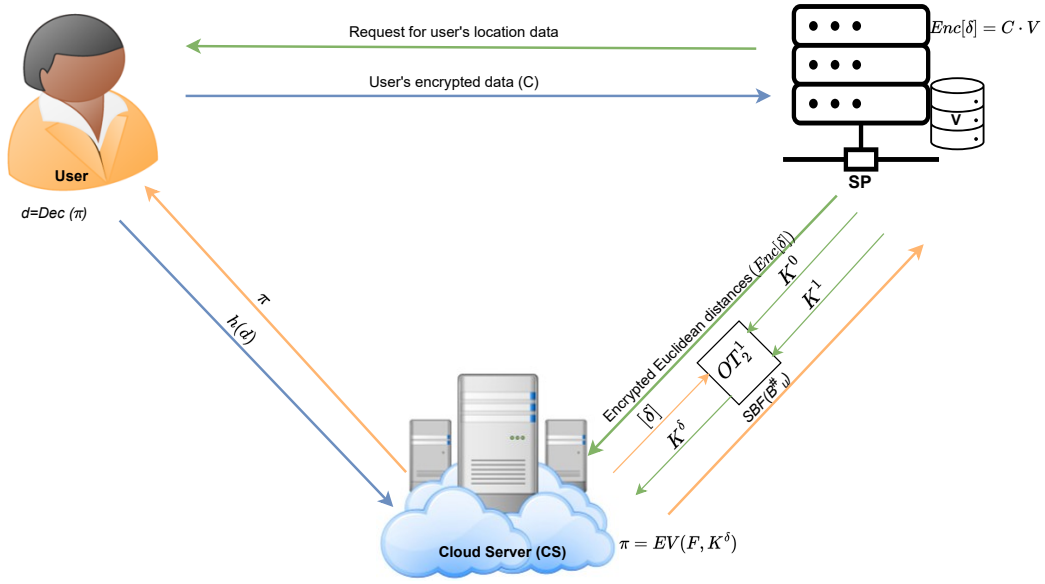


Figure 6.1: System Architecture for Secure Three-Party Computation. Most of the Heavy User-Side Computation Is Delegated to the Cloud Server (CS) Without Revealing the User's Private Input. Using the OT Protocol, the CS Can Securely Interact With the Service Provider (SP) in Order to Perform Secure Computations on the User's Behalf. The Arrow's Color Indicates the Flow of Information From One Party to Another, With Light Blue, Light Green, and Light Orange Representing the User, SP, and CS, Respectively.

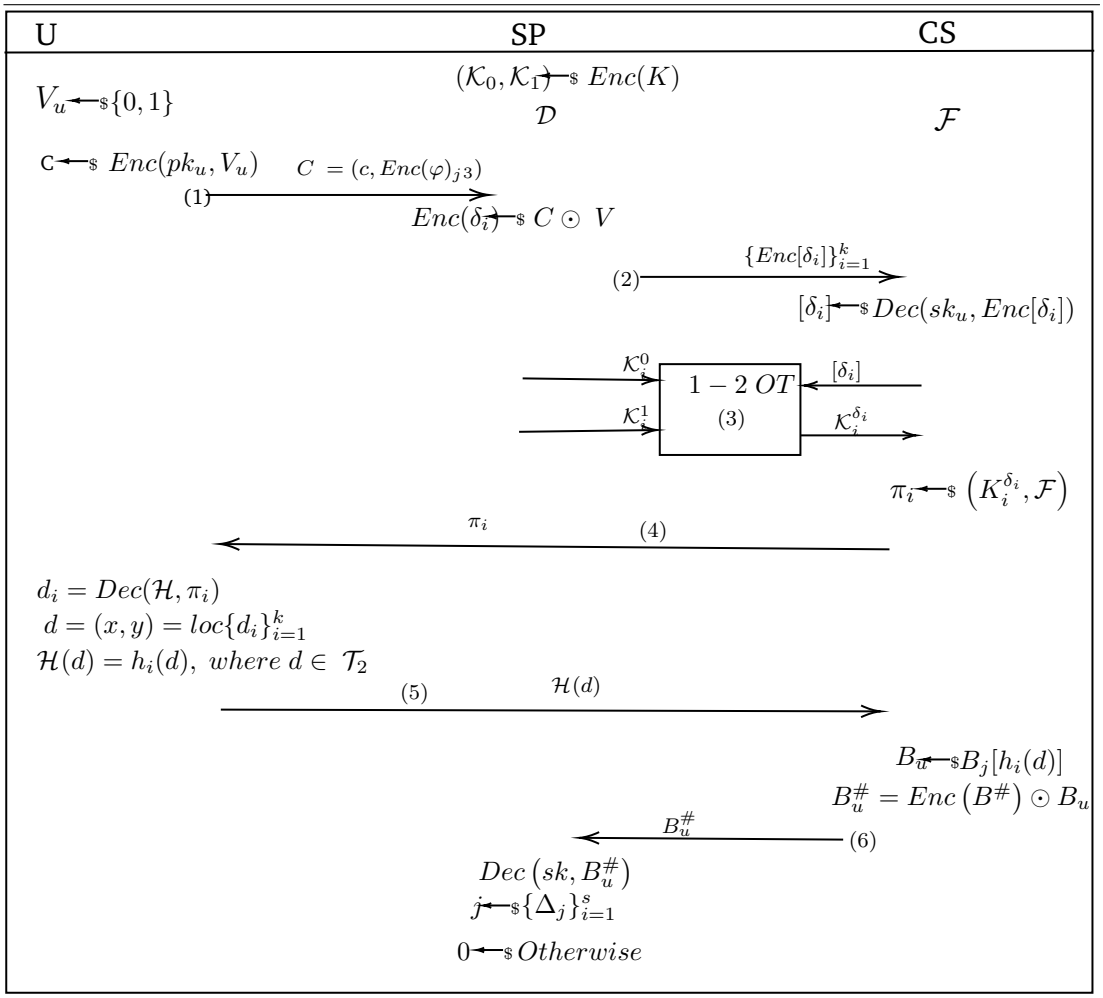
The protocol works as follows. First, the SP generates a public and private key pair (pk_s, sk_s) and constructs a spatial Bloom filter (SBF) $B^\#$ over the set of areas of interest $\xi = \{\triangle_1, \ldots, \triangle_l\}$ following the same process as in [CPM15]. Then, the SP encrypts the filter using its public key pk_s, and sends the encrypted filter $\mathsf{Enc}(B^\#)$ to the cloud server CS. This step can be done offline.
The user generates a public and private key pair (pk_u, sk_u). Then when the user wants to learn their location, they construct the fingerprint V_u of their location using RSS data and encrypt it using the public key pk_u, i.e., $c = \mathsf{Enc}\,(pk_u, V_u)$. The encrypted fingerprint c is sent to the SP, while the decryption key sk_u is forwarded to the cloud server CS on the assumption that the cloud server and the SP do not collude.
Using the client public key pk_u and the received fingerprint, the service provider SP can compute a masked encrypted Euclidean distance $\mathsf{Enc}[\delta]$ (by adding en-

crypted noise to the encrypted Euclidean distances $\mathsf{Enc}(\delta)$ to obscure it from the CS after decryption), using the reference database $\mathbb{D}$ (details of this are provided in the detailed protocol description).
The SP and the CS then engage in what is called a secure two-party computation using garbled circuit and oblivious transfer with two objectives: first, for the CS to remove the masking (noise) added to the encrypted Euclidean distances, $\mathsf{Enc}(\delta)$, by the service provider (SP); and secondly to compute the k-smallest Euclidean distances.
The CS sends the indexes of the k-nearest Euclidean distances (indistinguishable from random noise for the CS) to the user, who then retrieves the plain distance using the garbled output function. From the k nearest distances, the user localizes their exact location.
At regular intervals, as required by the application, the user determines the geocode (numerical representation of location coordinate) corresponding to their current location as described above. Subsequently, they encode the geocode using the hash functions sent by the service provider and send the hash digests to the CS. Finally, the CS constructs a filter (B_u) for the user using the received digests and performs the entry-wise homomorphic product of the user's filter (B_u) (plaintext) with that of the SP $(B^{\#})$ (ciphertext). The result of the operation, sent to the SP, allows it to determine the user's presence in an area of interest without revealing the area of interest to the user and the exact location to the service provider. The CS learns neither.
In the following, we provide a detailed description of the protocol, dividing it into different phases. The scheme requires an offline preparation and initialization phase, followed by four on-line phases: a measuring phase, a computation phase, a retrieval phase, and a server localization phase. The protocol in its entirety is depicted in Protocol 6. Algorithms 7-9 describe the protocol in pseudo-code from the point of view of the three parties: Algorithm 7 presents the client-side, Algorithm 8 the third party cloud server and Algorithm 9 the service provider. Specific steps in Protocol 6 are cross-referenced in Algorithms 7-9 for readability.

6.2.2.1 Preparation and Initialization Phase

In this phase, the SP publishes the localization-aided data $\{\mathcal{T}_1, \mathcal{T}_2\}$ of the reference database ($\mathbb{D}$) it holds to all users, where $\mathcal{T}_1 = \{ap_1, \cdots, ap_n\}$ is the access points and $\mathcal{T}_2 = \{\tau_1, \cdots, \tau_m\}$ represents the list of reference points. Importantly, it does not release the entire radio map. The user generates an encryption key

Protocol 6 Outsourcing Privacy-Preserving Protocol for Indoor Wi-Fi Localization



pair $< pk_u, sk_u >$ using the DGK cryptosystem and publishes the public key pk_u while the server uses an encryption scheme that allows private Hadamard product to generate key pairs $< pk_s, sk_s >$. Through a secure communication channel, the user sends the decryption key sk_u (this can always be revoked if compromised) to the CS. Using a keyed hash function $(\mathcal{H}^{\star})$, the SP constructs a spatial Bloom filter $(B^{\#})$ over the areas of interest $(\triangle)$ using the algorithm described in [CPM15] where $\triangle \in \xi$, and encrypts it with the public key (pk_s) to obtain $\mathsf{Enc}(B^{\#})$. It is worth noting, while ξ constitutes the set of all the areas of interest $(\triangle)$, $\xi \subseteq \mathcal{T}_2$, thus the exact location of the user, $d \in \mathcal{T}_2$ but may $\notin \xi$ depending on the position of the user. The service provider SP sends its public key (pk_s) together with the encrypted filter $\mathsf{Enc}(B^{\#})$ to the CS, and $\mathcal{H}^{\star}$ and garbled output function $[(\mathcal{H}(\pi^0), 0), (\mathcal{H}(\pi^1), 1)]$ to the user. The above process can be executed once and offline (that is, separately from the execution of the

Algorithm 7 User (U)

Input : $V_u = \{r_1 \cdots r_n\}; r_i \in \{0,1\}^n,\ \mathcal{H}$ **Output** : B_u

User Outsourced Location

7 : $C \leftarrow_\$ \emptyset$
8 : **for** $i \leftarrow_\$ 1 \cdots n$ **do**
9 : $c_i \leftarrow_\$ \mathsf{Enc}(-2r_i)$
10 : $\mathsf{Enc}(\varphi)_{j_3} \leftarrow_\$ \mathsf{Enc}(\sum_{i=1}^{n} r_i^2)$
11 : $C.append\{c, \mathsf{Enc}(\varphi)_{j_3}\}$
12 : **post** C *to* **S**

} *Step* (1) *of Protocol* 6

Retrieving Private Location

13 : $D \leftarrow_\$ \emptyset$
14 : **get** $\{\pi_i\}_{i=1}^{k}$ *from* **CS**
15 : $\{h_1, \cdots, h_k\} \leftarrow_\$ \mathcal{H}$
16 : **for** $j \leftarrow_\$ 1 \cdots k$ **do**
17 : $d_j \leftarrow_\$ lsb(\pi_j) \oplus h_j$
18 : **append**$\{d_j\}_j^k$ *to* D
19 : $d = (x, y) \leftarrow_\$ loc\{D\}$

} *Step* (4) *of Protocol* 6

remainder of the protocol).

6.2.2.2 Measuring Phase

The user generates the RSS fingerprint $V_u = \{r_1, \cdots, r_n\}$ of her position, and computes:

$$c = \mathsf{Enc}(-2r_1), \cdots, \mathsf{Enc}(-2r_n) \tag{6.3a}$$

$$\mathsf{Enc}(\varphi)_{j_3} = \mathsf{Enc}\Big(\sum_{i=1}^{n} r_{u_i}^2\Big)\ . \tag{6.3b}$$

The results are sent to the SP, as indicated in Algorithm 7 (step 1). The RSS measurement can always be done in an offline phase, and new RSS data is generated only when the user's position changes. Also, this is the only computation the user does before receiving the indexes of the k-smallest Euclidean distances from the CS. Interaction between the mobile user and the SP ends after the input query. The rest of the communication is either between the SP and the CS or the CS and the mobile user.

6.2.2.3 Computation Phase

After receiving the user's encrypted fingerprint $\{c_i\}_{i=1}^{n}$, the SP computes the masked encrypted Euclidean distances $\mathsf{Enc}[\delta]$ between the received fingerprint and the reference database $\mathbb{D}$ (Table 6.2). The results are forwarded to the CS. The computation is possible due to the homomorphic properties of the DGK cryptosystem [DGK07] as depicted in step (2) of Algorithm 9. Specifically, the SP computes:

$$\mathsf{Enc}(\varphi)_{j_1} = \mathsf{Enc}\Big(\textstyle\sum_{i=1}^{n} r_i^2\Big) \tag{6.4a}$$

$$\mathsf{Enc}(\varphi)_{j_2} = \textstyle\prod_{i=1}^{n}\Big(c_i\Big)^{r_i}, \tag{6.4b}$$

and the encrypted euclidean distance $\mathsf{Enc}(\delta)$ such that:

$$\begin{aligned}\mathsf{Enc}(\delta) &= \mathsf{Enc}\Big(\textstyle\sum_{i=1}^{n}(r_i - r_{u_i})^2\Big) \\ &= \mathsf{Enc}\Big(\sum_{i=1}^{n} r_i^2\Big) \cdot \textstyle\prod_{i=1}^{n}\Big(c_i\Big)^{r_i} \cdot \mathsf{Enc}\Big(\sum_{i=1}^{n} r_{u_i}^2\Big) \\ &= \mathsf{Enc}(\varphi)_{j_1} \cdot \mathsf{Enc}(\varphi)_{j_2} \cdot \mathsf{Enc}(\varphi)_{j_3}\end{aligned} \tag{6.5}$$

In order to protect the database $\mathbb{D}$ from being leaked to the cloud server or a malicious user (i.e., if the user colludes with the cloud server) , the service provider blinds (masks) each of the encrypted distances $\mathsf{Enc}(\delta_j)$ with a unique random value $\gamma_j \in \mathcal{R}$ such that:

$$\mathsf{Enc}\left[\delta_j\right] = \mathsf{Enc}(\delta_j) \cdot \mathsf{Enc}(\gamma_j) \tag{6.6}$$

where $\gamma_j \in \{0,1\}^n \ \forall \ j \ \in \{1, \cdots, m\}$. The SP finally sends the masked encrypted distances $\mathsf{Enc}\left[\delta\right]$ to the cloud server (CS) (line 34-38 of Algorithm 9). After receiving the masked encrypted distances $\{\mathsf{Enc}\left[\delta\right]_j\}_{j=1}^{m}$, the CS decrypts these distances using the user's decryption key sk_u to obtain unencrypted masked euclidean distances $\{[\delta]_j\}_{j=1}^{m}$ made up of the euclidean distances (δ) and the added noise (γ) as shown in equation 6.7. The process is described in Algorithm 8.

$$[\delta]_j = \delta_j + \gamma_j \tag{6.7}$$

In the first part, which is to remove the noise (γ), the SP and the CS use Yao's protocol (garbled circuit) and OT. The SP constructs a garbled circuit by first representing a subtraction function $a^{\hat{l}} - b^{\hat{l}} = [\delta] - \gamma$ as a Boolean circuit, where $\hat{l}$ is the bit-length of the distance. Substituting equation 6.7 into this function

yields:

$$a - b = [\delta] - \gamma = (\delta + \gamma) - \gamma = \delta \tag{6.8}$$

The subtraction circuit is similar to the one in [Sch12, KSS09], takes 2-input AND gate and 'free' XOR [KS08] computations (i.e., evaluation of the XOR-gates do not require any garble tables). The second part of the garbled function, $\mathcal{F}$, is to perform the comparisons & conditional swap using the technique of [JLL+19]. The entire circuit is converted into garbled circuit (encrypted circuit) and sent to CS. The SP's input values of the circuit $\gamma_1, \cdots, \gamma_m$, are converted into garbled values $\tilde{\gamma}_1, \cdots, \tilde{\gamma}_m$, which can be sent along with the circuit to CS. The garbled input of CS, $\left[\tilde{\delta}_1\right], \cdots, \left[\tilde{\delta}_m\right]$ corresponding to the masked distances $[\delta_1], \cdots, [\delta_m]$ are transferred to CS using parallel OT to enable the CS evaluates the circuit as shown in step (3) of Algorithm 8. At the end of the evaluation, CS obtains the indexes $\pi_1, \cdots, \pi_k$ corresponding to the k-smallest Euclidean distances $\delta_1, \cdots, \delta_k$. The CS then forwards the output indexes $\{\pi_i\}_{i=1}^{k}$ to the mobile user (line 27 of algorithm 6.3); these indexes are indistinguishable from random noise by the CS without the translation output table/function.

6.2.2.4 Retrieval Phase

When the user receives the garbled output indexes $\{\pi_i\}_{i=1}^{k}$ of the smallest k-nearest Euclidean distances, she uses the output function/translation table $[(\mathcal{H}(\pi^0), 0), (\mathcal{H}(\pi^1), 1)]$ sent by the SP to translate the garbled values (garbled output) to the corresponding plain indexes. The plain k-indexes correspond to the k-smallest euclidean distances d_j as depicted in step (4) of Algorithm 7. With the translation table, the user can determine the semantics of the output wire, i.e., $\mathcal{H}(\pi^0)$ represents 0 and $\mathcal{H}(\pi^1)$ represents 1 for all the $\hat{l}$ bits of each of the received euclidean distances. The user then estimates the private location d by computing the centroid of the k-nearest neighbours using $\mathcal{T}_2$ (the list of reference points).

6.2.2.5 Server Localization Phase

When the SP makes a request about the user's location, the user determines the geocode of $d(x_i, y_i) \in \mathcal{T}_2$. Using a keyed hash function $\mathcal{H}^\star$, the user computes the $\mathcal{H}^\star(d)$, i.e., the digest over the location d, and sends the results $\{h_1(d), \cdots, h_k(d)\}$ to the CS, where $h_i(d)$ is the hash values/digest over d.
When CS receives the hash values $\{h_i(d)\}_{i=1}^{k}$, creates a Bloom filter B_u, by map-

Algorithm 8 Cloud Server (CS)

Output : π

Obtaining User's Euclidean Distances

20 : **get** $\mathsf{Enc}[\delta]$ $from$ **SP**
21 : **for** $j \leftarrow\$\, 1 \cdots m$ **do**
22 : $[\delta_j] \leftarrow\$\, \mathsf{Dec}(sk_u, \mathsf{Enc}\,[\delta_j])$
23 : **return** $[\delta] \leftarrow\$\ [\delta_1] \cdots [\delta_m]$

1-2-Oblivious Transfer

24 : **get** $\left\{\mathcal{K}_i^0, \mathcal{K}_i^1\right\}_{i=1}^{m} \leftarrow\$\, \mathcal{K}$ $from$ **SP**
25 : $[\delta_1] \cdots [\delta_m] \leftarrow\$\, [\delta]$
26 : **for** $j \leftarrow\$\, 1 \cdots m$
27 : **foreach** $[\delta]$ **do**
28 : $\mathcal{K}_j^{\delta} \leftarrow\$\, \{\mathcal{K}_1^{\delta_1}, \cdots, \mathcal{K}_m^{\delta_m}\}$
29 : **return** $\mathcal{K}^{\delta} \leftarrow\$\, \mathcal{K}_1^{\delta} \cdots \mathcal{K}_m^{\delta}$

} *Step* (3) *of Protocol* 6

Circuit Evaluation

30 : **for** $j \leftarrow\$\, 1 \cdots k$ **do**
31 : $\pi_i \leftarrow\$\, \mathcal{F}(\mathcal{K}^{\delta})$
32 : **return** $\pi \leftarrow\$\, \pi_1 \cdots \pi_k$
33 : **post** π to **Client**

} *Step* (4) *of Protocol* 6

ping the hash values unto the filter and the number of $1s = \alpha$ counted, since this filter is binary and only takes 1s and 0s as input. Due to the multiplicative homomorphic properties, CS can compute $\mathsf{Enc}(B_u^{\#}) = \mathsf{Enc}(B^{\#}) \boxdot B_u$, using SP's public key (pk_s). If the user's current location was mapped unto to the spatial Bloom filter (SBF) by the SP, then after the multiplication of the two filters, all non-zero values in $\mathsf{Enc}(B_u^{\#})$ corresponding to the user's mappings in filter B_u, will remain, while all other positions in $\mathsf{Enc}(B_u^{\#})$ become zero. Finally, CS permutes the results, $\mathsf{Enc}(B_u^{\#})$, which changes the order to hide the location of the user, and sends both α and $\mathsf{Enc}(B_u^{\#})$ to the SP for verification.

When the SP receives the encrypted filter $\mathsf{Enc}(B_u^{\#})$, it decrypts it using the secret key sk_s, and subsequently performs location verification. If the number of non-zeros in the filter is $< \alpha$, then certainly the user is not within an area of interest i.e., $d \notin \triangle$. Else, if the number of non-zeros equals α, then the smallest output value $\triangle_j$, will be the area that contains the geocode of the user's location i.e., $(d \in \triangle_j)$, minus the false probability.

Algorithm 9 Service Provider (SP)

Input : $\{V_i\}_{i=1}^{n}, \ \gamma \xleftarrow{\mathcal{R}} \in \{0,1\}$ **Output** : $\mathsf{Enc}\,[\delta]$

User Outsourced Location

34 : **get** $c_i, \mathsf{Enc}(\varphi)_{j_3} \leftarrow\$\, C \ from$ **Client**
35 : **for** $j \leftarrow\$\, 1 \cdots m$ **do**
36 : **for** $i \leftarrow\$\, 1 \cdots n$ **do**
37 : $\mathsf{Enc}(\varphi)_{j_1} \leftarrow\$\, \mathsf{Enc}\left(\sum_{i=1}^{n} V_{ij}^2\right)$
38 : $\mathsf{Enc}(\varphi)_{j_2} \leftarrow\$\, \prod_{i=1}^{n} \left(c_i\right)^{v_{i,j}}$
39 : $\mathsf{Enc}(\delta_j) \leftarrow\$\, \mathsf{Enc}(\varphi)_{j_1} \cdot \mathsf{Enc}(\varphi)_{j_2} \cdot \mathsf{Enc}(\varphi)_{j_3}$
40 : **return** $\mathsf{Enc}(\delta) \leftarrow\$ \left\{\mathsf{Enc}(\delta_j)\right\}_{j=1}^{m}$

} $Step(2)\ of\ Protocol\ 6$

Masking the distances

41 : **for** $j \leftarrow\$\, 1 \cdots m$ **do**
42 : $\mathsf{Enc}(\gamma_j) \leftarrow\$\, \mathsf{Enc}(pk_s, \gamma_j)$
43 : $\mathsf{Enc}[\delta]_j \leftarrow\$\, \mathsf{Enc}(\delta_j) \cdot \mathsf{Enc}(\gamma_j)$
44 : **return** $\mathsf{Enc}[\delta] \leftarrow\$ \left\{\mathsf{Enc}[\delta]_j\right\}_{j=1}^{m}$
45 : **post** $\mathsf{Enc}[\delta]$ *to* **CS**

6.3 Security

The security of the proposed Protocol 6 relies on the underlying cryptographic primitives used. The security model is based on a malicious user and honest-but-curious non-colluding service provider (SP) and cloud server (CS). In other words, the SP and CS can analyze the data they receive to try to recover other parties' inputs but they should not deviate from the protocol and should not be under the same authority/administrator. The user can instead provide arbitrary inputs and deviate from the protocol. This is novel with respect to the leading solutions in the literature, including [EP19, LSZ+14], which offer protection only against *honest-but-curious* adversaries.

This security model, and in particular the non-collusion requirement, is established in the literature and is in line with current research on outsourced multi-party computation [CMTB16, JNO14, KMR12]. We argue that the model is also in line with industry practice and real-world expectations. While it is appropriate to assume a service provider and outsourced cloud server could become "curious", that is, learn as much as possible from their interactions, it is also re-

alistic to assume they do not actively deviate from the protocol. This, however, can not be assumed of a user. Non-collusion is also a common legal requirement in the industry sector, for instance, in the case of financial and technical independent audits [DRS10].
In the following, we first provide an informal overview of the privacy achieved by the protocol (Section 6.3.1); then we provide a formal security definition (Section 6.3.2) and proof (Section 6.3.3).

6.3.1 Privacy

Privacy in this protocol means:

- The SP only learns if the user is present in one of the limited predefined areas of interests $\triangle$ (and which area), but not the user's exact location. In other words, if the user is outside the predefined areas of interest, the SP learns nothing about the user's location. If the user is within an area, the SP learns the area only.
- The user only learns the estimated private location based on the user's input but nothing else. Thus, the SP reference database and the predefined areas $(\triangle)$ encoded in the filter are not disclosed to the user.
- The cloud server learns nothing and remains oblivious to both the user's location and the service provider's database.

The localization protocol starts when the mobile user first submits her input to the algorithm. This input (V_u) is always encrypted using her public key and the DGK-cryptosystem [DGK07]. Since the DGK cryptosystem is proven to be semantically secure (IND-CPA) under the hardness of the RSA subgroup assumption, it is impossible for the server to recover the plain RSS input of the user within a polynomial-time without the secret key (sk_u). Though the homomorphic properties of the DGK cryptosystem allow the server to compute k-nearest neighbours of the user, the SP cannot recover the plaintext (RSS values) from the k-nearest neighbours without first recovering the private key of the user sk_u. Even when the user happens to be in the same position twice or more, the probabilistic property of the DGK algorithm prevents the service provider from noticing due to randomness in the encryption of the input query. Even after the Euclidean distances have been computed, the semantic security of the DGK ciphertext remains intact. The protocol settings allow only the SP to construct the garbled circuit (GC), whose security is proven in [LP09]. The use of the OT as a tool for the GC prevents the SP from getting access to the CS's input $[\delta]$ of the user.

Notwithstanding that the CS has access to the user's private key sk_u, the CS cannot make any deduction from the masked Euclidean distances $[\delta]$ without access to the user's plain input. The security of GC further ensures that the gabled output π is indistinguishable from random values without the translation table. It was also shown in [BHR12] that GC is oblivious, such that given the garble function $\mathcal{F}$ and input $\mathcal{K}^{\delta}$ reveal nothing about the user's plain input. This preserves the privacy of the user's exact location from both the CS and SP. The user sends only the encoded hash values of her location to the CS. The caveat here is that this could potentially allow the CS to brute-force the location on the assumption that it has access to: the hash function, the geocode of $\mathcal{T}_2$ and knowledge of $\mathcal{T}_2$. We argue this is unfeasible without colluding with SP. Nevertheless, the use of the keyed hash function prevents this attack and thereby guarantees the privacy of the user's location even if the CS has knowledge of ξ or $\mathcal{T}_2$.

Mapping the hash values $\{h_1(d), \cdots, h_k(d)\}$ into the filter by CS ensures that the user's exact location (longitude and latitude coordinates) are never leaked to the SP. Considering that cryptographic hash functions are one-way functions, it is infeasible to reverse the operations. Besides, the shuffling of the filter $\mathsf{Enc}(B_u^{\#})$ hides the user's exact location should the user be in any of the predefined areas of interest $(\triangle)$. When the SP decrypts the filter $\mathsf{Enc}(B_u^{\#})$, and the number of non-zero values in $(B_u^{\#})$ is equal to α, it implies the user is within one of the areas of interest $(\triangle)$ with the area being the smallest positive integer minus false positive. It is statistically infeasible for the SP to infer from the filter $(B_u^{\#})$ the user's exact location due to the re-ordering of the non-zero values. Equally, if the number of non-zero values is less than α, then surely the SP learns nothing about the user's location which in effect preserves the user's privacy.

From the SP perspective, the user only receives output based on her input. The user does not participate in the computation after submitting her encrypted input; hence, there is little a malicious user can do to corrupt the computation of her location. The GC circuit outputs only the indexes (π), of the smallest k-nearest Euclidean distances (δ) to the CS. Thus, if the GC is semantically secure (IND-CPA), then the protocol leaks no information about SP's database $(\mathbb{D})$ because the garbled values of the circuit are indistinguishable from pseudo-random noise. We hold the same security definition as [BHR12], that GC is private because having access to the decoding function and the output values (indexes of the k smallest distances) leaks no information about the database

and the user's input. In the worst-case scenario, arbitrary inputs by the user can only affect the correctness of the protocol but nothing else. Moreover, blinding each of the Euclidean distances with different random values $\mathcal{R}$ prevents the CS from learning any information about $\mathbb{D}$ after decryption, assuming it colludes with the user. The security of the oblivious protocol (OT) ensures that the CS receives only outputs that correspond to its inputs (decrypted masked Euclidean distances $[\delta]$) without gaining any additional information about the SP's input. Therefore, the privacy of the SP's database ($\mathbb{D}$) is preserved since the output of the protocol to the user is only the indexes of the k-smallest neighbours.
The use of homomorphic encryption (which is statistically indistinguishable) to encrypt the filter preserves the privacy of the predefined areas encoded in the filter. The homomorphic properties allow CS to compute private multiplication of $\mathsf{Enc}(B^{\#})$ and B_u without gaining any additional information about the encoded filter $\mathsf{Enc}(B_u^{\#})$.
Though an adversary may attempt to build their own database ($\tilde{\mathbb{D}}$), similar to $\mathbb{D}$ over the same area, this would not give them an advantage in learning the SP areas of interest (δ). Furthermore, constructing an alternative $\tilde{\mathbb{D}}$ is either impracticable in the given setting or useless: if the indoor space has restricted areas, the attack is impracticable as the user does not have access to those restricted areas (considering a critical infrastructure such as an airport), and we can assume even the accessible indoor areas in question are a controlled environment. Attacks on the physical infrastructure are outside the scope of this work. If, instead, the indoor space is publicly accessible and information on the location of the AP is public as well, the user does not gain any further knowledge in constructing $\tilde{\mathbb{D}}$, and the SP's areas of interest are never revealed to the user. Finally, a database ($\tilde{\mathbb{D}}$), built by an adversary, is unlikely to make the same accurate prediction of the SP's well-trained reference database ($\mathbb{D}$).

6.3.2 Security Definition

We formulate a security definition based on the Real/Ideal world paradigm similar to [Gol04, KMR11]. As indicated earlier, our model setting involves 3-parties in the execution of the protocol. One of the parties (the user, such as a mobile client) can be malicious. The other two servers (SP and CS) are independent non-colluding parties. In this setting, each party's input is simulated in an ideal view of the execution of the protocol. And, if the view during the simulation in the ideal world is computationally indistinguishable from that of the real execution of the protocol (Λ) based on the party's input/output, then the

protocol is considered secure. In other words, no party learns any additional information except what the protocol intended to output. Furthermore, our secure multiparty computations protocol has only one-output function $f(x_1, x_2)$ based on the user's and SP's input. We now define the Ideal/Real-world execution of the protocol below:

Real-world execution: The execution of the protocol (Λ) in the real-world model involves parties $P = \{U, CS, SP\}$ and a subset corresponding adversary $A = \{A_U, A_{CS}, A_{SP}\}$ who can corrupt the associated party. Let x_i be the protocol input, and $b_i \in_R \{0,1\}^*$ be the auxiliary input (a string or side information that an attacker may have prior to the protocol's execution) and k the security parameter during the execution of the protocol. Except for the CS, who only inputs b, all parties input x and b. The two servers (SP and CS) are semi-honest and non-colluding, according to the model assumption. As a result, both servers will correctly input the appropriate data. This condition, however, does not apply to the malicious user. Let OUT^i be the output for the party $A_i \in A$, and OUT^j for $i \neq j$, be the output of the honest party during the execution of the protocol. Thus, the protocol output during the real execution (*real view*) of the protocol is defined as:

$$REAL_{\Lambda}^{View}[A](\bar{x}, k)_{k\in\mathbb{N}} \stackrel{\text{def}}{=} OUT_{\Lambda}^{i} \cup \{OUT_{\Lambda}^{j} : j \in H\}$$

where $\bar{x} = \{x_1, x_2, b_1, b_2, b_3\}$, and H is honest parties.

Ideal-world execution: The ideal world of the protocol execution has the same settings as the Real-world model with the same parties and input. However, in the ideal world, a trusted party receives all input from the parties participating in the protocol. The trusted party (TTP) securely computes the output function based on the received parties' input. Thus let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$, represent the ideal functionality where:

Inputs: x_i denotes the input of User (U) and Service provider,(SP). Let b_i be the auxilliary input of adversary A and security parameter k for all party (U, CS ,SP) including TTP.

The honest party (semi-honest) P_j always sends its input $(x_i' = x)$ as required to the trusted party (TTP). However, the corrupted party A_i (malicious) can decide whether to send its arbitrary input (x') or not to the TTP. Should the corrupted party decide not to send its input, an empty string $(\perp)$ or abort message is sent to the trusted party. In that case, an abort message is delivered to all parties. Else, the execution of the protocol continues, i.e., the trusted party (TTP) evaluates the function f.

Output: Let OUT^j be the output $(f(x'_1, x'_2))$, returned to the honest party $H \subseteq P$ by the trusted party (TTP) and OUT^i be the arbitrary output (computed within probalistic polynomial time) returned to the corrupted party A_i. Thus the i-th partial output of the *ideal-world* execution of f between the parties (U, CS, SP) in the presence of adversary A (independently malicious simulators) is given by :

$$IDEAL_f^{View}[A](\bar{x}, k)_{k\in\mathbb{N}} \stackrel{\text{def}}{=} OUT_f^i \ \cup \{OUT_f^j : j \in H\}$$

We now provide a security definition for the three-party protocol based on the introduction of the Ideal/Real-world protocol executions.

Definition 6.3.1 (Security)**.** *A three-party protocol Λ, securely computes the function f (with an abort option in the presence of malicious parties), if for every probabilistic polynomial time (PPT) adversary A, for the real model, $\exists$ a PPT adversary S in the ideal model, such that :*

$$\left\{IDEAL_f^{View}[S, A](\bar{x}, k) \ \stackrel{\text{c}}{\approx} \ REAL_\Lambda^{View}[A](\bar{x}, k)\right\}_{k\in\mathbb{N}}$$

where $\stackrel{\text{c}}{\approx}$ $means$ computationally indistinguishable.

Our privacy-preserving protocol for indoor Wi-Fi-localization is secure according to Theorem 6.3.2 below and meets the security Definition 6.3.1. The proof can be found in the following (Section 6.3.3).

Theorem 6.3.2. *Protocol 6 computes the function f securely in the presence of a malicious user, a semi-honest SP, and a semi-honest CS.*

6.3.3 Security Proof

In showing the protocol's security, we formulate the view of the real execution of the protocol by constructing the view of the simulator for each considered adversary (A_U, A_{CS}, A_{SP}) based on the input/output in the ideal world. If the view of the ideal world is computationally indistinguishable from the view in the real execution of the protocol, then we conclude that the protocol is secure. Thus, the adversary does not learn any extra information apart from what the protocol intended to output.

View of a malicious User: From the client perspective, its input to the protocol is the encryption of r_i, and the output is the garbled values of the k-nearest Euclidean distances. We choose a simulator S_U, which runs A_U, and controls the other parties. Similar to the real world, A_U obtains r_i and sends it to S_U.

As the adversary can make any arbitrary input, if A_U sends an incorrect input bitlength or an empty string, the experiment terminates. Else, S_U simulates the user's view by encrypting r_i since its has the keys . Since the experiment controls both the SP and the CS, S_U computes the garbled values $\tilde{\gamma}_i$ and $\tilde{\delta}_i$ like in the real execution of the protocol Λ. S_U then queries the trusted party TP for the evaluation of $f(\tilde{\gamma}_i, [\tilde{\delta}_i])$. Finally, S_U sends to A_U the garbled output of the k-nearest Euclidean distances and the output translation function $[(\mathcal{H}(\pi^0), 0), (\mathcal{H}(\pi^1), 1)]$ as in the real execution of the protocol. Due to the semantic security of the garbled output (random values), the view of the adversary A_U in the ideal world is indistinguishable from the view of the real-world execution of the protocol; that is, the ideal/real output of the protocol are computationally indistinguishable.

View of the (semi-honest) Service Provider: Now we construct a simulator S_{SP} for in the ideal world to simulate the adversary's (A_{SP}) view in a real execution of the protocol. In this scenario, S_{SP} runs A_{SP} and controls the other parties. In the ideal execution, the simulator S_{SP} prepares its input of choice, similar to the mobile user's input (r_i), in the real execution of the protocol. This input is encrypted and sent to the A_{SP} . The adversary A_{SP} now performs its computation similar to the steps in the real execution of the protocol and computes the garbled circuit as well. A_{SP} then sends to S_{SP} the garbled circuits alongside its garbled input $\tilde{\gamma}_i$. Since the adversary A_{SP} is semi-honest as in the definition of the security model, the right input is expected to be sent else the experiments terminate.

Since the simulator controls the other parties, A_{SP} and the simulator engages in 1-2 OT protocol to deliver the keys corresponding to the simulator's input $([\delta_i])$ as in the real execution of the protocol. The simulator S_{SP}, now having both input to the function f, queries the trusted party TP, to evaluate $f(\tilde{\gamma}_i, [\tilde{\delta}_i])$. Due to the security of oblivious transfer, it's evident that the adversary A_{SP} cannot learn any information about the simulators' input of the garbled circuit. In addition, the semantic security (IND-CPA) of the DGK cryptosystem makes it impossible for A_{SP} to learn any meaningful information about the Mobile user's input within a polynomial time similar to the real execution of the protocol. Furthermore, any modification or incorrect construction of the circuit will cause the experiment to terminate; bearing in mind that the server is a semi-honest party, this shouldn't be a problem. Thus, the adversary A_{SP}, has no way to statistically distinguish the above interaction in the ideal world from that of the real execution of the protocol.

View of the (semi-honest) Cloud Server: Finally, we construct a simulator

(S_{CP}) which in the ideal world can simulate the adversary's view (A_{CP}) in the real execution of the protocol.
The output of CS is the l-bits of the smallest Euclidean distances (garbled output). In this experiment, the simulator S_{CS} runs A_{CS} and controls the other parties (SP and the user). First S_{CP}, sends to A_{CP} an encrypted masked distance (Enc$[\delta]$). This allows A_{CP} to decrypts the encrypted masked distance as in the real-execution of the protocol. The simulator S_{CP} then simulates a garbled circuit and sends it to A_{CS} alongside the server's garbled input $(\tilde{\gamma}_i)$. Finally, S_{CP} simulates an OT protocol and engages A_{CP} in a 1-2 oblivious transfer protocol to allow A_{CS} to recover the keys to its garbled input. A_{CP} then evaluates the circuit and sends back to S_{CP} garbled output values of Euclidean distances. Since the A_{CP} is semi-honest per the definition of the security model, it will not tamper with the garbled output, else that will cause S_{CP} to trigger $(\perp)$, guaranteeing the correct output of the garbled circuit. Also, the semantic security (IND-CPA) of the garbled output and the security of the Oblivious transfer protocol makes the garbled values indistinguishable from pseudo-random noise. As a result, the A_{CS} is unable to distinguish between the simulated interaction in the ideal world and the real execution of the protocol.

6.3.4 Potential Expansion to a Malicious Service Provider

Although this protocol was designed to provide security against a semi-honest service provider, in this section, we introduce and briefly discuss some potential capabilities when the service provider is assumed to be malicious. This might be useful for the reader in comprehending the reasoning behind the choice of the security model for the protocol construction. As a malicious adversary, the most likely scenario for the SP is to construct the Spatial Bloom filter so that verifying the user's location is always true. In other words, the SP will include all locations of the infrastructure in question so that the user's location is always disclosed, regardless of whether the user is within an area of interest. However, we believe that additional hardening of the protocol could easily reveal that the SP is cheating, as the size of the areas of interest could be agreed upon beforehand, revealing the size of the filter; thus, any additional insertion into the filter will result in the SP being caught, as the size of the filter will exceed what was agreed upon. Also, as SP is the constructor of the garbled circuit, in order to send the garbled version of $\mathbf{F}$, SP can easily send the incorrect F to CP, as SP is not required to adhere to the protocol's instructions. As CS only receives ciphertext or encrypted circuits, it is unable to confirm that the

correct $\mathbf{F}$ or circuit was sent. This intent will affect the protocol's correctness. In reality, this is a problem of both correctness and privacy. Thus, the output of the evaluation may appear to be random garbage, but it reveals the CS's entire input (user's location). A potential solution to address the issue of a malicious party constructing a circuit is to use the cut-and-choose paradigm. This approach involves the constructor creating multiple copies of the circuits and the evaluator requesting the constructor to open more than half of them. By doing so, if the constructor has constructed the circuit incorrectly, there is a high probability of being caught during the verification process. This approach provides a way to ensure the security and accuracy of the circuit construction in the presence of malicious parties.

6.4 Evaluation

This work makes use of homomorphic encryption to preserve the privacy of the users' queries. Using homomorphism clearly increases the computation and communication incurred on the user side, which is why in the design of the protocol, we outsourced most of the user's computations. Due to the higher bandwidth and computational capabilities, we can safely assume to be at the disposal of the SP and CS, the cost analysis focuses more on the user side, which we presume to be a constrained mobile or IoT device. For completeness, we nonetheless include the SP and CS computation and communication costs in Section 6.4.3.

For this analysis, we use the recommendations [NIS16, ENC18] of key size up-to-long-term protection. The security parameters; T denotes the asymmetric key and t for the symmetric key. Though a key-length $T = 1024$ (in Table 6.5) is widely acknowledged to be obsolete for security by the cryptographic community, we included it for the sake of completeness of the analysis.

In the following, we evaluate the computational and communication complexity of the proposed cryptographic protocol, and we compare it to the current leading solutions in the literature. In particular, we compare it directly to [LSZ$^+$14] and [EP19], as these schemes have comparable constructions. As neither of these schemes has been implemented, we compare the computational complexity in terms of exponentiations and multiplications as standard in the relevant literature. In both cases, the proposed protocol has a significant complexity advantage. This stands even if the improvements to previous protocols proposed in [YJ18] for [LSZ$^+$14] are included.

Table 6.3: Comparison of Computational Complexity at Client-Side.

Protocol	Preprocessing	Retrieval
[LSZ+14]	$2n\ Exp + 4n\ Mul$	$m\ Exp$
[EP19]	$2n\ Exp + 4n\ Mul$	$m\ Exp$
This protocol	$(n+1)\ Exp$	$negligible$

6.4.1 Computational Complexity

Analyzing computational complexity is essential to secure multi-party computation (SMC) efficiency, especially considering IoT devices.

Localization protocols using additively homomorphic encryption mostly use Paillier encryption [Pai99], where the hardness relies on the composite residual problem with RSA modulus N as the public key. This protocol instead uses DGK [DGK07, DGK09], which has smaller plaintext space but with better performances. We use the DGK for encryption, while the GC is for the subtraction and computing of the shortest Euclidean distances.

To make fair comparisons with other similar localization protocols, we use the optimized version of these protocols, where most of the expensive modular exponentiation is pushed to the set-up phase. Table 6.3depicts a summary of the findings. In protocols [LSZ+14, EP19], like in the previous chapter, a total of $2n$ elements (RSS) of the user's input query are encrypted using the Paillier encryption. The cost of Paillier encryption requires two modular exponentiation (Exp) and one modular multiplication (Mul). The complexity of the encryption is improved by choosing $g = N + 1$ in the equation 2.2 to become:

$$\mathsf{Enc}(x) = (xN + 1) \cdot r^N \mod N^2 \tag{6.9}$$

making the encryption complexity to be only one exponentiation and two modular multiplications instead. Thus in total, each user requires $2n$ exponentiation and $4n$ modular multiplications for each input query. Using the DGK cryptosystem results in the encryption complexity of one exponentiation modulo N being more efficient than the Paillier due to the lower exponent. In total, the client requires approximately n exponentiation, which is a significant improvement. In [LSZ+14, EP19], decrypting one ciphertext requires one exponentiation with optimization technique while using our outsourced protocol in this chapter requires negligible computation.

Comparing this protocol to [YJ18], which suggested the use of Paillier and GC as a solution to the security weakness in [LSZ+14], our proposed protocol has better performance, especially the heavy GC computations at the user side are

Table 6.4: Time Required by the User to Perform the Required Encryption Computation for Different Key Sizes. The Table Details the Mean Encryption Times in Milliseconds (ms) Averaged Over 10 Encryption Rounds. This Illustrates the Relationship Between Increasing Key Size and Encryption Duration, Highlighting the Computational Impact of Stronger Encryption in Secure Communications.

Key Size (bits)	**Encryption Time (ms)**
1024	0.14
2048	0.38
3072	0.79
4096	1.26

all delegated to the CS, making it more practicable in a real-world setting. Even when most of the computations are in the setup phase, clients in [YJ18] in the online phase still need to compute n exponentiation for n ciphertexts and GC evaluation requiring lm non-XOR gates for subtraction. Evaluating the GC for the indexes of k smallest distances by the client using the k nearest neighbor search algorithm of [SHSK15], and later improved by [JLL+19] would still require the invocation of cryptographic hash functions of $k(2ml + m\log_2 m)$ non-XOR gates for k comparisons and $2k$ conditional swap, where $\log_2 m$ is the circuit depth for m values of bit-length $\hat{l}$. This computation is undoubtedly expensive for a mobile device (especially IoT) with limited resources to perform compared to the proposed protocol.

To assess the computational burden on the user-side, particularly for devices with constrained processing power, we provide experimental results on the time required to perform the encryptions needed to securely transmit data to the servers. The analysis focuses on the user, as SP and CS are presumed to be equipped with robust computational resources. The implementation was done using Python, and run on a 2.40 GHz Intel(R) Core(TM) processor. The results, illustrated in Table 6.4, comprise the time complexity for both equations 6.3a and 6.3b, which are encryptions performed by the user before transmitting data to the Service provider (SP). Specifically, we use a sample $r_i = \{-40, -42, -46, -49, -50, -55\}$, obtained from an experiment on live access points. This evaluation is crucial for understanding the practicality and efficiency of the protocol in real-world scenarios, especially when used by devices with constrained resources. The results highlight the relatively small computational effort and time required, which would be easily achieved on any modern user device (such as a smartphone).

Table 6.5: Comparison of Bandwidth Usage at Client-side: For This Protocol, the Client's Traffic Emanates From the Cloud Server.

	Preprocessing			Online		
Security Level	Related work		This Protocol	Related work		This Protocol
	[LSZ$^+$14]	[EP19]		[LSZ$^+$14]	[EP19]	
Asymptotic	$4nT$	$4nT$	$(n+1)T$	$2mT$	$2mT$	$k\hat{l}$
1024	6.1 kB	6.1 kB	1.7 kB	236.5 kB	236.5 kB	0.015 kB
2048	12.3 kB	12.3 kB	3.3 kB	473.1 kB	473.1 kB	0.015 kB
3072	18.4 kB	18.4 kB	5.0 kB	709.6 kB	709.6 kB	0.015 kB
15360	92.1 kB	92.1 kB	25.0 kB	3548.2 kB	3548.2 kB	0.015 kB

6.4.2 Communication Complexity

The communication complexity of Paillier and DGK is largely dependent on the size T of the RSA modulus N. The size of ciphertext of Paillier cryptosystem is as twice as that of DGK. Moreover, unlike protocol [EP19, LSZ$^+$14], where $2n$ output ciphertexts are sent from the client to the SP, the client in this protocol sends a total of $n+1$ ciphertexts to the SP, an improvement by a factor of approximately 2 in the online communication.

Table 6.5 shows a significant gain in this protocol's communication bandwidth compared to that of [EP19, LSZ$^+$14]. The user's encrypted message sent to the SP in this protocol has a communication bit of $(n+1)T$ asymptotically compared to $4nT$ of [EP19, LSZ$^+$14]. As shown in Table 6.5, as the keysize increases, the gain in bandwidth in our protocol asymptotically approaches a factor of 4.

The encrypted distances that the SP transfers to the client in [EP19, LSZ$^+$14] require an asymptotic communication overhead of $2mT$ bits; juxtaposing it, this protocol has a much lower communication complexity of kl bits. Table 6.5 depicts the pattern of gains for different security parameters. Let $m = 924$ and $n = 520$ in the database ($\mathbb{D}$) as in [TMU$^+$14]. The scan result at each reference point depends on the internal structure and the positions of the installed access points. [LSZ$^+$14] estimated an average of 6 APs is enough to localize a user. To estimate the communication in bandwidth, we set $n = 12$[1] *goal* in table 6.5 for the average scan result per reference point that a user sends to the SP. And let $k = 5$ and maximum of $\hat{l} = 24$, (where $\hat{l} = log_2 n + 2l\ bits + 1$, is the maximum bit-length of the euclidean distance). During the online phase, as the key-size of [EP19, LSZ$^+$14] increases, the bandwidth size grows proportionally while in our proposed protocol, the bandwidth remains constant.

[1]The work of [LSZ$^+$14] showed that scan results (number of available ap) ≥ 6 per reference point is enough to localize a client. In [TMU$^+$14], the average number of available AP per reference point is 18 while in [LT14], it ranges from 11 to 67.

The estimated communication cost in the suggested solution [YJ18] in the client/server settings still poses a significant communication overhead at the client side. The communication cost for sending one Paillier ciphertext will be $2T$. The OT protocol of [AIR01] against a semi-honest SP and malicious user will result in communication complexity bits of $\approx 12mlt$. And the offline communication complexity for transferring the GC will be lm non-XOR gates for the subtraction and approximately $3klm$ (klm for comparisons and $2klm$ for the minimum selection) for computing the Euclidean distances. Thus, using the Garbled row reduction of [NPS99] will result in a complexity of approximately $3t$ bits per non-XOR gate.
Therefore, this outsourced protocol brings significant communication bandwidth gains, considering the limited resources at the client side. Though combining the efficient OT protocol of [Bea95] and [NP01] will allow most of the expensive operations to be done in the set-up phase, and in effect, improve the complexity. Nevertheless, the bandwidth involved would still be an overburden for a resource constraint client in addition to the direct impact on power consumption. Without the outsourcing server, the total online communication complexity in the phase of optimization would asymptotically be $2mT$ for m Paillier ciphertexts transferred by the SP plus $2m\hat{l}t$ of the online part of the OT protocol for m values of bit-length $\hat{l}$ and security parameter t. Thus communication complexity of $k\hat{l} - bits$ for k values of bit-length $\hat{l}$ in this protocol for resource constraint mobile client is a significant improvement over [YJ18], not to mention the communication overhead in the set-up phase for the client.

6.4.3 Communication and Computation for SP and CS

Our focus for the cost analysis is mainly on the user side, who normally uses a constrained mobile device and is the party that benefits the most from a reduction in overheads. However, for completeness, we briefly discuss the approximate computational complexity of the service provider (SP) and cloud server (CS), as well as their communication costs. The results are summarised in Table 6.6, where $\mathcal{L} = 2^l - 1$ and l is the bit-length of the RSS value. To accurately track the cost of communication between a Service Provider (SP) and a Cloud Server (CS) in a secure computing environment, it is necessary to consider all relevant factors. Therefore, the communication Table 6.6 records the full cost of this communication, which includes the DGK ciphertext, the total ciphertext sent in the Oblivious Transfer (OT) step, and the garbled values received by the CS. This differs from the first Table 6.5, which merely monitors the total

number of packets the client receives from the Cloud server. Considering these variables, the communication table provides a more comprehensive and realistic depiction of communication costs in a safe computing environment. It is important to remember that both parties (CS and SP) are very likely to have no constrain in communication capabilities and a very high computational capacity. Nonetheless, our results show that both computation and communication overheads are perfectly manageable even by low-resource servers.

In the set-up phase, a total of *mn* exponentiation is needed to compute $\mathsf{Enc}(v^2)$ plus m encryption of the noise (γ). For the garbled circuit, SP will invoke 2^2 hash functions for each 2-input non-xor gate created; the technique of [ZRE15] will bring it down to only 2 per 2-input non-xor gate. The random oracle (RO) protocol of [NP01] will result in $2\hat{l}$ invocation of cryptographic hash functions for the computation of OT by the SP in the set-up phase. In computing the filter, the SP will invoke approximately k cryptographic hash functions for each position inserted in the filter. The CS, on the other hand, only needs one invocation of the cryptographic hash function for each 2-input non-xor gate. It is important to highlight that most SP's computations would be more cost-effective than shown in the theoretical analysis, as most RSS values are zero at each reference point. Given that cloud service providers have state-of-the-art infrastructure and tremendous resources at their disposal, computational work by the CS is easily doable. Computations for all non-xor gates are negligible.

In the online phase, the cost for computing $\mathsf{Enc}(\varphi)_{j_2}$ will be $\approx \mathcal{L}m$ exponentiations (where $\mathcal{L} = 2^l - 1$ and l is the bit-length of the RSS value) plus *mn* modular multiplications. Finally, $2m$ modular multiplications are required in computing $\mathsf{Enc}(\delta)$ and m modular multiplication for computing $\mathsf{Enc}[\delta]$.

With regards to communication complexity, the SP will send a total of $3\hat{l}t$ per each 2-input non-XOR gate using the garbled row reduction [NPS99] for the garbled circuit plus $\approx 6\hat{l}t$ for the OT based on the random oracle [NP01] . In the online phase, the communication cost will be asymptotically mT of DGK ciphertext and $3\hat{l}mt$ *bits* ($\hat{l}mt$ for SP's garbled input and $2\hat{l}mt$ for the OT based on two rounds of communication between SP and CS)

Table 6.6: Computational and Communication Complexity at the SP and CS.

		Preprocessing			Online		
		DGK	GC	OT			
Computation	SP	$mn\ Exp$	$2kt(2m\hat{l} + m\log_2 m)$	$2\hat{l}$	$\mathcal{L}m\ Exp + m(n+3)Mul + km$		
	CS		$kt(2m\hat{l} + m\log_2 m)$	$\hat{l}$	$m\ Exp + 1\ Mul$		
Communication			$3\hat{l}mt$	$6\hat{l}mt$	DGK mT	GC $\hat{l}mt$	OT $2m\hat{l}t$

6.5 Conclusion

In this chapter, we introduce an efficient privacy-preserving protocol for indoor Wi-Fi localization that ensures privacy for both service providers and users. Our innovative approach combines cryptographic primitives such as additive homomorphic encryption, garbled circuits, and spatial Bloom filters, which maintain privacy while delivering accurate location output. The proposed protocol significantly reduces computational costs and online communication, making it suitable for IoT environments. Notably, our solution protects against malicious users, setting it apart from similar protocols. Refer to Chapter 7.1.3 for a complete discussion and conclusion.

Chapter 7

Conclusions & Future Work

7.1 Conclusions

This thesis sheds light on the ever-increasing popularity of location-based services (LBS) as well as the advantages those tailored services bring to users. However, it also shed light on the significance of location privacy as an essential component of our day-to-day lives, demonstrating that it is not only essential for businesses to uphold compliance with regulations such as GDPR to avoid the associated fines but also has the potential to have a detrimental effect on the person who is being targeted. Therefore, the right to one's locational privacy has become an intrinsic entitlement that must always be protected. The sensitive information that is acquired from location data is often unique and can be exploited or sold to third parties without the consent of the users.
This information can indicate a person's religious belief owing to the frequent visitation of religiously linked structures or show your purchasing behaviour due to the stores that the user typically visits. In fact, the type of medical facility or consulting practice that a person typically utilises can give some indication regarding the health status of the targeted user. In addition, the individual's insurance premiums could be quite expensive if the insurance company had access to the restaurants that the individual frequents or their driving behaviour through location data.
As location-based services evolve, new challenges and privacy risks will undoubtedly emerge. Therefore, the work presented in this thesis should be considered part of an ongoing effort to develop and refine techniques that preserve user privacy while maintaining the functionality and efficiency of LBS. The exploration of alternative cryptographic primitives, the improvement of existing protocols, and the development of new privacy-preserving methods will be es-

sential to ensuring the continued growth and success of location-based services in a secure and privacy-conscious manner.
This thesis has demonstrated the effectiveness of the extent to which this sensitive information can be revealed and proposed a novel, well-designed, privacy-preserving protocol that can effectively prevent any privacy leaks, particularly in light of the growing popularity and utility of LBS. The presented solutions primarily concentrated on the indoor Wi-Fi environment to make the most of the existing Wi-Fi infrastructure. These solutions have also been published in top-tier peer-reviewed journals and conferences hosted by ACM and IEEE.

7.1.1 De-anonymization Attacks on Real-World Location Dataset

In chapter 4, we proposed and implemented a de-anonymization model for geolocation data. To de-anonymize the target users, our attack employs the hidden Markov model (by taking into account Spatio-temporal trajectories) to create user mobility profiles (user fingerprinting).
We build the HMM by first predicting the users' initial areas of interest or frequently visited locations (known as hidden states) and the days (observations) on which users visited these locations before feeding into the model. This is achieved using a machine learning technique like the DBSCAN clustering algorithm to predict the hidden states from the target geolocation data.
We assess the robustness of attack techniques based on ground truth mobility trajectories from two different cities (Shanghai and Rome). Overall, our proposed de-anonymization model is able to effectively re-identify anonymized users by leveraging the power of HMM to model user mobility patterns even in the presence of noise or missing data. Based on the evaluation score, our attack techniques successfully re-identify up to 85% and 90% anonymized users in the GeoLife and taxicab datasets, respectively. This represents a significant improvement over existing de-anonymization attack techniques in the literature, and has potential applications in various settings where privacy and security are essential considerations.
One potential drawback associated with the proposed de-anonymization model is the fact that it is dependent on the availability of a training dataset in order to make accurate predictions regarding the mobility patterns of the target datasets or users. In some instances, it may be challenging or impossible to access such a dataset, especially now that a lot of datasets are being classified as private or

closed-sourced. In such circumstances, the performance of our model may be significantly reduced.
Our model also has the drawback of assuming that the target users' mobility trajectories are stationary over time. In other words, users' areas of interest or frequently visited locations do not vary much across the entirety of the dataset. This assumption may only hold in some cases and could lead to reduced performance if users' mobility patterns change significantly over time.

7.1.2 A Privacy-Preserving Protocol for Indoor Wi-Fi Localization

Chapter 5 makes use of secure multi-party computations (SMC) as one potential solution for addressing privacy concerns regarding de-anonymising location datasets or protecting users in mobility datasets. SMC enables multiple parties to jointly compute functions on their private inputs without disclosing their inputs in order to preserve their privacy. In the context of mobility datasets, service providers may share location datasets with third parties or among themselves without violating the users' privacy. It is particularly useful in situations in which the disclosure of location data could reveal sensitive information, as well as those in which the parties involved distrust one another or have competing interests.
This thesis proposes a two-party privacy-preserving protocol for indoor Wi-Fi localization, which addresses the privacy concerns of both the user and the service provider in light of the aforementioned advantage of SMC and its many other advantages. In particular, our scheme makes use of partial homomorphic encryption, also known as Paillier's cryptosystem. This encryption method ensures the user's location remains private by performing the localization computation in the encrypted domain. This, in turn, ensures that most of the computational overhead on the user side is delegated to the server while hiding the exact location from the server. The use of homomorphic encryption also preserves the data privacy of the service provider (SP). We use spatial Bloom filters in addition to homomorphic encryption enabling the SP to use location-based service to learn the predefined areas of the user, but not the exact location, while preventing the user from learning these predefined areas. When the user is outside these predefined areas in the building, the service provider learns nothing about the user's location.
We finally discuss the security of the protocol considering semi-honest parties,

i.e. they follow the instructions of the protocol but try to gain additional knowledge. For future work, we plan to extend the security features of the protocol to defend against malicious adversaries. A three-party model can be researched to outsource most of the computation on the user side, which is useful in constrained embedded systems, especially in defending against malicious adversaries. In a three-party scenario, further security guarantees for both the user and the server in the presence of a third party would need to be researched.

7.1.3 A Cryptographic Protocol for Efficient Mutual Location Privacy Through Outsourcing in Indoor Wi-Fi Localization

Chapter 6 proposed a new privacy-preserving protocol for indoor Wi-Fi localization that addresses the limitations of the indoor Wi-Fi localization protocol introduced in chapter 5. Instead of the assumption of a *semi-honest* adversary presented in chapter 5, the new protocol incorporates a more realistic security assumption in the form of a *malicious* adversary, which is one of the most important improvements. This is particularly important because of the diverse and unpredictable IoT environment. As a result, designing a privacy-preserving protocol capable of withstanding the possible threat posed by malicious adversaries or actors is vital. In order to achieve such a security assumption as well as improved communication and computation complexity, our protocol uses a combination of cryptographic primitives as a building block to protecting the privacy of the parties involved. In particular, our scheme employs additive homomorphic encryption (DGK encryption) to preserve the privacy of the user's location fingerprint while allowing the service provider to make computations over the encrypted fingerprint. Whereas the use of a garbled circuit preserves the privacy of the SP's reference database against malicious users, and at the same time, delivers the location output (k-nearest neighbours) to the user. The spatial Bloom filter, another important privacy-preserving feature of the proposed protocol allows the SP to use the location-based service in the indoor environment to learn the user's vicinity in predefined areas of interest while preventing the user from learning these predefined areas. When the user happens to be outside these predefined areas of interest ($\triangle$), the service provider learns nothing about the client's location.
Compared to other similar protocols [EP19, LSZ+14], our protocol's complexity analysis shows a very significant reduction in computational costs at the user-

side, as most of the heavy computations of the garbled circuit operations are securely and effectively outsourced to the cloud server. Online communication costs show a reduction of about 99.99% compared to other similar protocols in the client/server setting, making our protocol more efficient and practicable in the Internet of Things environment. Additionally, our protocol is the first among those analyzed to provide security against malicious users, with other protocols limited to honest-but-curious adversaries.

7.2 Future Work

In chapter 4, we proposed a de-anonymization algorithm for mobility trajectories that outperformed existing de-anonymization techniques in the literature. However, even with this performance, there are limitations to this proposed algorithm that need future work to address. For instance, the performance of the algorithm is dependent on the availability of the training dataset used to generate the mobility behaviour. Future work could address this limitation, particularly in situations where mobility trajectories are limited. One possible solution could be using an unsupervised machine learning algorithm that can effectively learn the target users' mobility behaviour from the limited dataset without the need for training the dataset or manually annotating the dataset before using a supervised machine learning algorithm. Another area of direction for future work could be the development techniques that can adapt our model to handle non-stationary user mobility trajectories. This is particularly important when the mobility behaviour of users in a mobility dataset changes quickly. One solution to handle such a dataset could be using a more sophisticated machine learning algorithm, like recurrent neural networks (RNNs), to model the evolution of users' mobility behaviour over time.

In chapter 6, a privacy-preserving protocol against malicious users was proposed. As the security guarantees that are provided by this protocol only assume that service providers are semi-honest, it will be worthwhile to investigate potential directions for future work in further strengthening such protection against actively corrupt service providers or cloud services. This could be achieved by implementing additional cryptographic techniques. One toolkit that can be leveraged on could be the ABY [DSZ15] framework for efficient mixed-protocol multiparty computation. Therefore, using the ABY framework will not only improve efficiency but will also improve the protocol's scalability, making it more practically applicable for use in an environment involving the Internet

of Things.

Furthermore, the current setup of the protocol makes use of a variety of cryptographic primitives as a building block to preserve the privacy of the parties. However, it will be worthwhile to consider other protocols or cryptographic primitives that improve not only the efficiency of the proposed protocol but also further improve its security and privacy. These improvements could come in the form of a combination of the two. This could involve techniques that will optimize the current efficiency of the protocol, such as reducing the size of the garbled circuit. For instance, utilizing secret sharing, which has a lower communication overhead, could be a very good option, particularly when dealing with low-latency networks.

Acronyms

AoI Area of Interest.
AP Access Point.
COPPA Children's Online Privacy Protection Act.
DBSCAN Density-Based Spatial Clustering of Applications with Noise.
DGK Damgård, Geisler and Krøigaard.
FHE Fully Homomorphic Encryption.
FTC Federal Trade Commission.
GDPR General Data Protection Regulation.
GLONASS Global Navigation Satellite System.
GNSS Global Navigation Satellite System.
GPS Global Positioning System.
GSM Global System for Mobile communication.
HIPPA Health Insurance Portability and Ac- countability Act.
HMM Hidden Markov Model.
IND-CPA INDistinguishability under Chosen Plaintext Attack.
IOT Internet of Things.
IPS Indoor Positioning System.
LBS Location-Based Services.
LSP Location Service Provider.
NYT New York Times.
OT Oblivious Transfer.
PII personally identifiable information.
POI Points of Interest.
RSA Rivest-Shamir-Adleman.
RSS Received Signal Strength.
SBF Spatial Bloom filter.
SMC Secure Multi-party Computation.
SP Service Providers.
SSN Social Security Numbers.

TVM Temporal Vector Map.
Wi-Fi Wireless Fidelity.

References

[ACD+07] Claudio A. Ardagna, Marco Cremonini, Ernesto Damiani, Sabrina De Capitani di Vimercati, and Pierangela Samarati. Location privacy protection through obfuscation-based techniques. In Steve Barker and Gail-Joon Ahn, editors, *Data and Applications Security XXI, 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security, Redondo Beach, CA, USA, July 8-11, 2007, Proceedings*, volume 4602 of *Lecture Notes in Computer Science*, pages 47–60. Springer, 2007.

[AFL+16] Toshinori Araki, Jun Furukawa, Yehuda Lindell, Ariel Nof, and Kazuma Ohara. High-throughput semi-honest secure three-party computation with an honest majority. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 805–817. ACM, 2016.

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *EUROCRYPT 2001, Intl. Conf. on the Theory and Application of Cryptographic Techniques*, pages 119–135, 2001.

[Ama06] Amazon. Amazon web services, 2006.

[AMS+15] Dalal Al-Azizy, David E. Millard, Iraklis Symeonidis, Kieron O'Hara, and Nigel Shadbolt. A literature survey and classifications on data deanonymisation. In Costas Lambrinoudakis and Alban Gabillon, editors, *Risks and Security of Internet and Systems - 10th International Conference, CRiSIS 2015, Mytilene, Lesbos Island, Greece, July 20-22, 2015, Revised Selected Papers*, volume 9572 of *Lecture Notes in Computer Science*, pages 36–51. Springer, 2015.

[ASC+17] Amr Al-Anwar, Yasser Shoukry, Supriyo Chakraborty, Bharathan

Balaji, Paul Martin, Paulo Tabuada, and Mani B. Srivastava. Proloc: resilient localization with private observers using partial homomorphic encryption: demo abstract. In Pei Zhang, Prabal Dutta, and Guoliang Xing, editors, *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN 2017, Pittsburgh, PA, USA, April 18-21, 2017*, pages 257–258. ACM, 2017.

[BB05] Mauro Brunato and Roberto Battiti. Statistical learning theory for location fingerprinting in wireless lans. *Computer Networks*, 47(6):825–845, 2005.

[BBL+] Lorenzo Bracciale, Marco Bonola, Pierpaolo Loreti, Giuseppe Bianchi, Raul Amici, and Antonello Rabuffi. Crawdad dataset roma/taxi (v. 2014-07-17), traceset: taxicabs.

[Bea95] Donald Beaver. Precomputing oblivious transfer. In *CRYPTO '95, 15th Annual Intl. Cryptology Conf.*, pages 97–109, 1995.

[BHR12] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In *ACM Conf. on Computer and Communications Security, CCS'12*, pages 784–796, 2012.

[Blo70] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

[BMR90] Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols (extended abstract). In *22nd Annual ACM Symposium on Theory of Computing*, pages 503–513, 1990.

[BP66] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[BP00] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: an in-building rf-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000, The Conference on Computer Communications, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Reaching the Promised Land of Communications, Tel Aviv, Israel, March 26-30, 2000* [DBL00], pages 775–784.

[BS68] Leonard E Baum and George Sell. Growth transformations for functions on manifolds. *Pacific Journal of Mathematics*, 27(2):211–227, 1968.

[CBM20] Luca Calderoni, Samantha Bandini, and Dario Maio. Privacy preservation in outsourced mobility traces through compact data structures. *J. Inf. Secur. Appl.*, 55:102630, 2020.

[CDN15] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[CFFM15] Luca Calderoni, Matteo Ferrara, Annalisa Franco, and Dario Maio. Indoor localization in a hospital environment using random forest classifiers. *Expert Syst. Appl.*, 42(1):125–134, 2015.

[CHY+18] Jing Chen, Kun He, Quan Yuan, Min Chen, Ruiying Du, and Yang Xiang. Blind filtering at third parties: An efficient privacy-preserving framework for location-based services. *IEEE Trans. Mob. Comput.*, 17(11):2524–2535, 2018.

[CK17] Kyung Choi and Mihui Kim. Locap: Privacy-preserving location proximity protocol. In *Ninth International Conference on Ubiquitous and Future Networks, ICUFN 2017, Milan, Italy, July 4-7, 2017*, pages 994–998. IEEE, 2017.

[CMP12] Luca Calderoni, Dario Maio, and Paolo Palmieri. Location-aware mobile services for a smart city: Design, implementation and deployment. *J. Theor. Appl. Electron. Commer. Res.*, 7(3):74–87, 2012.

[CMTB16] Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin R. B. Butler. Secure outsourced garbled circuit evaluation for mobile devices. *Journal of Computer Security*, 24(2):137–180, 2016.

[CPM15] Luca Calderoni, Paolo Palmieri, and Dario Maio. Location privacy without mutual trust: The spatial bloom filter. *Computer Communications*, 68:4–16, 2015.

[CPM18] Luca Calderoni, Paolo Palmieri, and Dario Maio. Probabilistic properties of the spatial bloom filters and their relevance to cryptographic protocols. *IEEE Trans. Information Forensics and Security*, 13(7):1710–1721, 2018.

[DBL00] *Proceedings IEEE INFOCOM 2000, The Conference on Computer Communications, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Reaching the Promised Land of Communications, Tel Aviv, Israel, March 26-30, 2000*. IEEE, 2000.

[DBL04] *Proceedings IEEE INFOCOM 2004, The 23rd Annual Joint Confer-*

ence of the IEEE Computer and Communications Societies, Hong Kong, China, March 7-11, 2004. IEEE, 2004.

[DBL08] *INFOCOM 2008. 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 13-18 April 2008, Phoenix, AZ, USA*. IEEE, 2008.

[DBL11] *32nd IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA*. IEEE Computer Society, 2011.

[DBL14] *2014 IEEE Conference on Computer Communications, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014*. IEEE, 2014.

[DCC09] Raibatak Das, Christopher W. Cairo, and Daniel Coombs. A hidden markov model for single particle tracks quantifies dynamic interactions between LFA-1 and the actin cytoskeleton. *PLoS Comput. Biol.*, 5(11), 2009.

[DGK07] Ivan Damgård, Martin Geisler, and Mikkel Krøigaard. Efficient and secure comparison for on-line auctions. In *Information Security and Privacy, 12th Australasian Conf., ACISP 2007*, pages 416–430, 2007.

[DGK09] Ivan Damgård, Martin Geisler, and Mikkel Krøigaard. A correction to 'efficient and secure comparison for on-line auctions'. *IJACT*, 1(4):323–324, 2009.

[DK05] Matt Duckham and Lars Kulik. A formal model of obfuscation and negotiation for location privacy. In Hans-Werner Gellersen, Roy Want, and Albrecht Schmidt, editors, *Pervasive Computing, Third International Conference, PERVASIVE 2005, Munich, Germany, May 8-13, 2005, Proceedings*, volume 3468 of *Lecture Notes in Computer Science*, pages 152–170. Springer, 2005.

[DMHVB13a] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3(1):1–5, 2013.

[dMHVB13b] Yves-Alexandre de Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, 3, 2013.

[DMRSP15] Yves-Alexandre De Montjoye, Laura Radaelli, Vivek Kumar Singh, and Alex "Sandy" Pentland. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*,

347(6221):536–539, 2015.

[DRS10] Frank Dölitzscher, Christoph Reich, and Anthony Sulistio. Designing cloud services adhering to government privacy laws. In *10th IEEE Intl. Conf. on Computer and Information Technology, CIT 2010*, pages 930–935. IEEE Computer Society, 2010.

[DSZ15] Daniel Demmler, Thomas Schneider, and Michael Zohner. ABY - A framework for efficient mixed-protocol secure two-party computation. In *22nd Annual Network and Distributed System Security Symposium, NDSS 2015*. The Internet Society, 2015.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad, editors, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pages 226–231. AAAI Press, 1996.

[ENC18] ENCRYPT. Algorithms, key size and protocols report (2018), h2020-ict-2014 – project 645421, d5.4, ecrypt-csa, 2018.

[EP19] Samuel N. Eshun and Paolo Palmieri. A privacy-preserving protocol for indoor wi-fi localization. In *16th ACM Intl. Conf. on Computing Frontiers, CF 2019*, pages 380–385, 2019.

[EP22] Samuel N. Eshun and Paolo Palmieri. Two de-anonymization attacks on real-world location data based on a hidden markov model. In *IEEE European Symposium on Security and Privacy, EuroS&P 2022 - Workshops, Genoa, Italy, June 6-10, 2022*, pages 1–9. IEEE, 2022.

[EP24] Samuel N. Eshun and Paolo Palmieri. A cryptographic protocol for efficient mutual location privacy through outsourcing in indoor wi-fi localization. *IEEE Transactions on Information Forensics and Security*, pages 1–1, 2024.

[EU16] GDPR EU. General data protection regulation, 2016.

[FHdM21] Ali Farzanehfar, Florimond Houssiau, and Yves-Alexandre de Montjoye. The risk of re-identification remains high even in country-scale location datasets. *Patterns*, 2(3):100204, 2021.

[FKZ+20] Marco Fiore, Panagiota Katsikouli, Elli Zavou, Mathieu Cunche, Françoise Fessant, Dominique Le Hello, Ulrich Matchi Aïvodji, Baptiste Olivier, Tony Quertier, and Razvan Stanica. Privacy in trajectory micro-data publishing: a survey. *Trans. Data Priv.*,

13(2):91–149, 2020.

[FLC18] Yunhe Feng, Zheng Lu, and Qing Cao. Secure sharing of private locations through homomorphic bloom filters. In *4th IEEE International Conference on Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing, and IEEE International Conference on Intelligent Data and Security, BigDataSecurity/HPSC/IDS 2018, Omaha, NE, USA, May 3-5, 2018*, pages 107–113. IEEE, 2018.

[FS14] Kassem Fawaz and Kang G. Shin. Location privacy protection for smartphone users. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 239–250. ACM, 2014.

[FSH11] Julien Freudiger, Reza Shokri, and Jean-Pierre Hubaux. Evaluating the privacy risk of location-based services. In George Danezis, editor, *Financial Cryptography and Data Security - 15th International Conference, FC 2011, Gros Islet, St. Lucia, February 28 - March 4, 2011, Revised Selected Papers*, volume 7035 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2011.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178. ACM, 2009.

[GKdPC14] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. De-anonymization attack on geolocated data. *J. Comput. Syst. Sci.*, 80(8):1597–1614, 2014.

[GKKB09] Gabriel Ghinita, Panos Kalnis, Murat Kantarcioglu, and Elisa Bertino. A hybrid technique for private location-based queries with database protection. In Nikos Mamoulis, Thomas Seidl, Torben Bach Pedersen, Kristian Torp, and Ira Assent, editors, *Advances in Spatial and Temporal Databases, 11th International Symposium, SSTD 2009, Aalborg, Denmark, July 8-10, 2009, Proceedings*, volume 5644 of *Lecture Notes in Computer Science*, pages 98–116. Springer, 2009.

[GKKB11] Gabriel Ghinita, Panos Kalnis, Murat Kantarcioglu, and Elisa Bertino. Approximate and exact hybrid algorithms for private nearest-neighbor queries with database protection. *GeoInformat-*

ica, 15(4):699–726, 2011.

[GLP+13] Oana Goga, Howard Lei, Sree Hari Krishnan Parthasarathi, Gerald Friedland, Robin Sommer, and Renata Teixeira. Exploiting innocuous activity for correlating users across sites. In Daniel Schwabe, Virgílio A. F. Almeida, Hartmut Glaser, Ricardo Baeza-Yates, and Sue B. Moon, editors, *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 447–458. International World Wide Web Conferences Steering Committee / ACM, 2013.

[GM02] Ayse Göker and Hans I. Myrhaug. User context and personalisation. In *6th European Conf. on Case Based Reasoning, ECCBR 2002*, pages 1–7, 2002.

[Gol04] Oded Goldreich. *The Foundations of Cryptography - Volume 2: Basic Applications*. Cambridge University Press, 2004.

[Gro04] Litum Group. Employee tracking rtls system, 2004.

[GVR20] Location intelligence market size, share & trends analysis report. Technical Report GVR-2-68038-401-7, Grand View Research, Feb 2020.

[GWS05] Hans-Werner Gellersen, Roy Want, and Albrecht Schmidt, editors. *Pervasive Computing, Third International Conference, PERVASIVE 2005, Munich, Germany, May 8-13, 2005, Proceedings*, volume 3468 of *Lecture Notes in Computer Science*. Springer, 2005.

[HWH+18] Qin Hu, Shengling Wang, Chunqiang Hu, Jian-Hui Huang, Wei Li, and Xiuzhen Cheng. Messages in a concealed bottle: Achieving query content privacy with accurate location-based services. *IEEE Trans. Veh. Technol.*, 67(8):7698–7711, 2018.

[IEE16] IEEE. Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. In *Proceedings IEEE INFOCOM IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2016*, 2016.

[JKS+18] Kimmo Järvinen, Ágnes Kiss, Thomas Schneider, Oleksandr

Tkachenko, and Zheng Yang. Faster privacy-preserving location proximity schemes. In Jan Camenisch and Panos Papadimitratos, editors, *Cryptology and Network Security - 17th International Conference, CANS 2018, Naples, Italy, September 30 - October 3, 2018, Proceedings*, volume 11124 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2018.

[JLL+19] Kimmo Järvinen, Helena Leppäkoski, Elena Simona Lohan, Philipp Richter, Thomas Schneider, Oleksandr Tkachenko, and Zheng Yang. PILOT: practical privacy-preserving indoor localization using outsourcing. In *IEEE European Symp. on Security and Privacy, EuroS&P 2019*, pages 448–463, 2019.

[JNO14] Thomas P. Jakobsen, Jesper Buus Nielsen, and Claudio Orlandi. A framework for outsourcing of secure computation. In Gail-Joon Ahn, Alina Oprea, and Reihaneh Safavi-Naini, editors, *6th edition of the ACM Workshop on Cloud Computing Security, CCSW '14*, pages 81–92. ACM, 2014.

[JR85] Biing-Hwang Juang and Lawrence R. Rabiner. A probabilistic distance measure for hidden markov models. *AT&T Tech. J.*, 64(2):391–408, 1985.

[JZF+15] Pei Jiang, Yunzhou Zhang, Wenyan Fu, Huiyu Liu, and Xiaolin Su. Indoor mobile localization based on wi-fi fingerprint's important access point. *IJDSN*, 11:429104:1–429104:8, 2015.

[KCZ+15] Andreas Konstantinidis, Georgios Chatzimilioudis, Demetrios Zeinalipour-Yazti, Paschalis Mpeis, Nikos Pelekis, and Yannis Theodoridis. Privacy-preserving indoor localization on smartphones. *IEEE Trans. Knowl. Data Eng.*, 27(11):3042–3055, 2015.

[KHdMR20] Dániel Kondor, Behrooz Hashemian, Yves-Alexandre de Montjoye, and Carlo Ratti. Towards matching user mobility traces in large-scale datasets. *IEEE Trans. Big Data*, 6(4):714–726, 2020.

[KK04] Kamol Kaemarungsi and Prashant Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting. In *Proceedings IEEE INFOCOM 2004, The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, Hong Kong, China, March 7-11, 2004* [DBL04], pages 1012–1022.

[KL51] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[KMR11] Seny Kamara, Payman Mohassel, and Mariana Raykova. Outsourcing multi-party computation. *IACR Cryptol. ePrint Arch.*, page 272, 2011.

[KMR12] Seny Kamara, Payman Mohassel, and Ben Riva. Salus: a system for server-aided secure function evaluation. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *ACM Conf. on Computer and Communications Security, CCS'12*, pages 797–808. ACM, 2012.

[KMRS14] Seny Kamara, Payman Mohassel, Mariana Raykova, and Seyed Saeed Sadeghian. Scaling private set intersection to billion-element sets. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, volume 8437 of *Lecture Notes in Computer Science*, pages 195–215. Springer, 2014.

[KN21] Jon Keegan and Alfred Ng. There's a multibillion-dollar market for your phone's location data. *The Markup*, Sep 2021.

[Kru07] John Krumm. Inference attacks on location tracks. In Anthony LaMarca, Marc Langheinrich, and Khai N. Truong, editors, *Pervasive Computing, 5th International Conference, PERVASIVE 2007, Toronto, Canada, May 13-16, 2007, Proceedings*, volume 4480 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2007.

[KS08] Vladimir Kolesnikov and Thomas Schneider. Improved garbled circuit: Free XOR gates and applications. In *Automata, Languages and Programming, 35th Intl. Colloquium, ICALP 2008, Proc., Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, pages 486–498, 2008.

[KS14] Hiroaki Kikuchi and Jun Sakuma. Bloom filter bootstrap: Privacy-preserving estimation of the size of an intersection. *JIP*, 22(2):388–400, 2014.

[KSS09] Vladimir Kolesnikov, Ahmad-Reza Sadeghi, and Thomas Schneider. Improved garbled circuit building blocks and applications to auctions and computing minima. In *Cryptology and Network Security, 8th Intl. Conf., CANS 2009*, pages 1–20, 2009.

[LCC+05] Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian E. Smith, James Scott, Timothy Sohn, James

Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello, and Bill N. Schilit. Place lab: Device positioning using radio beacons in the wild. In Gellersen et al. [GWS05], pages 116–133.

[LLS16] Shujun Li, Hong Li, and Limin Sun. Privacy-preserving crowdsourced site survey in wifi fingerprint-based localization. *EURASIP J. Wirel. Commun. Netw.*, 2016:123, 2016.

[LNV11] Kristin E. Lauter, Michael Naehrig, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? *IACR Cryptol. ePrint Arch.*, page 405, 2011.

[LP09] Yehuda Lindell and Benny Pinkas. A proof of security of yao's protocol for two-party computation. *J. Cryptology*, 22(2):161–188, 2009.

[LSZ+14] Hong Li, Limin Sun, Haojin Zhu, Xiang Lu, and Xiuzhen Cheng. Achieving privacy preservation in wifi fingerprint-based localization. In *2014 IEEE Conf. on Computer Communications, INFOCOM 2014*, pages 2337–2345, 2014.

[LT14] E. Lohan and J. Talvitie. Indoor wlan measurement data, 2014.

[LTK11] Elina Laitinen, Jukka Talvitie, and Prashant Krishnamurthy. Comparison of positioning accuracy of grid and path loss-based mobile positioning methods using receiver signal strengths. In *Proceedings SPAMEC 2011, Signal Processing and Applied Mathematics for Electronics and Communications, August 26-28, 2011*, pages 1–4, 2011.

[LXB+21] Pintu Lohar, Guodong Xie, Malika Bendechache, Rob Brennan, Edoardo Celeste, Ramona Trestian, and Irina Tal. Irish attitudes toward COVID tracker app & privacy: Sentiment analysis on twitter and survey data. In Delphine Reinhardt and Tilo Müller, editors, *ARES 2021: The 16th International Conference on Availability, Reliability and Security, Vienna, Austria, August 17-20, 2021*, pages 37:1–37:8. ACM, 2021.

[LYZ15] Dongdai Lin, Moti Yung, and Jianying Zhou, editors. *Information Security and Cryptology - 10th International Conference, Inscrypt 2014, Beijing, China, December 13-15, 2014, Revised Selected Papers*, volume 8957 of *Lecture Notes in Computer Science*. Springer, 2015.

[LZW21] Xingxin Li, Youwen Zhu, and Jian Wang. Highly efficient

privacy preserving location-based services with enhanced one-round blind filter. *IEEE Trans. Emerg. Top. Comput.*, 9(4):1803–1814, 2021.

[MDBP08] Yoni De Mulder, George Danezis, Lejla Batina, and Bart Preneel. Identification via location-profiling in GSM networks. In Vijay Atluri and Marianne Winslett, editors, *Proceedings of the 2008 ACM Workshop on Privacy in the Electronic Society, WPES 2008, Alexandria, VA, USA, October 27, 2008*, pages 23–32. ACM, 2008.

[Mic10] Microsoft. Microsoft azure, 2010.

[Mil07] Elizabeth Million. The hadamard product. *Course Notes*, 3(6), 2007.

[MKH17] Takao Murakami, Atsunori Kanemura, and Hideitsu Hino. Group sparsity tensor factorization for re-identification of open mobility traces. *IEEE Trans. Inf. Forensics Secur.*, 12(3):689–704, 2017.

[MNPS04] Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella. Fairplay - secure two-party computation system. In *13th USENIX Security Symposium*, pages 287–302, 2004.

[MTC08] Liran Ma, Amin Y. Teymorian, and Xiuzhen Cheng. A hybrid rogue access point protection framework for commodity wi-fi networks. In *INFOCOM 2008. 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 13-18 April 2008, Phoenix, AZ, USA* [DBL08], pages 1220–1228.

[MYYR10] Chris Y. T. Ma, David K. Y. Yau, Nung Kwan Yip, and Nageswara S. V. Rao. Privacy vulnerability of published anonymous mobility traces. In Nitin H. Vaidya, Suman Banerjee, and Dina Katabi, editors, *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking, MOBICOM 2010, Chicago, Illinois, USA, September 20-24, 2010*, pages 185–196. ACM, 2010.

[NIS16] NIST. Recommendation for key management, special publication 800-57 part 1 rev. 4, 2016.

[NLZ+14] Ben Niu, Qinghua Li, Xiaoyan Zhu, Guohong Cao, and Hui Li. Achieving k-anonymity in privacy-aware location-based services. In *2014 IEEE Conference on Computer Communications, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014*, pages 754–762. IEEE, 2014.

[NLZ+15] Ben Niu, Qinghua Li, Xiaoyan Zhu, Guohong Cao, and Hui Li. Enhancing privacy through caching in location-based services. In *2015 IEEE Conference on Computer Communications, INFOCOM 2015, Kowloon, Hong Kong, April 26 - May 1, 2015*, pages 1017–1025. IEEE, 2015.

[NP99] Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 573–590. Springer, 1999.

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *Proc. of the Twelfth Annual Symposium on Discrete Algorithms*, pages 448–457, 2001.

[NP05] Moni Naor and Benny Pinkas. Computationally secure oblivious transfer. *J. Cryptology*, 18(1):1–35, 2005.

[NPS99] Moni Naor, Benny Pinkas, and Reuban Sumner. Privacy preserving auctions and mechanism design. In *First ACM Conf. on Electronic Commerce (EC-99)*, pages 129–139, 1999.

[NS08] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (S&P 2008), 18-21 May 2008, Oakland, California, USA*, pages 111–125. IEEE Computer Society, 2008.

[NUTV16] Farid Movahedi Naini, Jayakrishnan Unnikrishnan, Patrick Thiran, and Martin Vetterli. Where you are is who you are: User identification by matching statistics. *IEEE Trans. Inf. Forensics Secur.*, 11(2):358–372, 2016.

[OPS22] Ivan Oleynikov, Elena Pagnin, and Andrei Sabelfeld. Outsourcing MPC precomputation for location privacy. In *IEEE European Symposium on Security and Privacy, EuroS&P 2022 - Workshops, Genoa, Italy, June 6-10, 2022*, pages 504–513. IEEE, 2022.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT '99, Intl. Conf. on the Theory and Application of Cryptographic Techniques*, pages 223–238, 1999.

[PCM14] Paolo Palmieri, Luca Calderoni, and Dario Maio. Spatial bloom filters: Enabling privacy in location-aware applications. In *Infor-*

mation Security and Cryptology - 10th Intl. Conf., Inscrypt 2014, pages 16–36, 2014.

[PH78] Stephen C. Pohlig and Martin E. Hellman. An improved algorithm for computing logarithms over gf(p) and its cryptographic significance (corresp.). *IEEE Trans. Information Theory*, 24(1):106–110, 1978.

[PKYB14] Russell Paulet, Md. Golam Kaosar, Xun Yi, and Elisa Bertino. Privacy-preserving and content-protecting location based queries. *IEEE Trans. Knowl. Data Eng.*, 26(5):1200–1210, 2014.

[PSDG09] Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. Crawdad data set epfl/mobility (v. 2009-02-24), 2009.

[PSSW09] Benny Pinkas, Thomas Schneider, Nigel P. Smart, and Stephen C. Williams. Secure two-party computation is practical. In *ASIACRYPT 2009, 15th Intl. Conf. on the Theory and Application of Cryptology and Information Security*, pages 250–267, 2009.

[Rab81] M Rabin. How to exchange secrets by oblivious transfer. harvard aiken comp. lab. Technical report, TR-81, 1981.

[Rab89] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[RHWL22] Sai Deepika Regani, Yuqian Hu, Beibei Wang, and K. J. Ray Liu. Wifi-based robust indoor localization for daily activity monitoring. In *MWSSH@MobiCom 2022: Proceedings of the 1st ACM Workshop on Mobile and Wireless Sensing for Smart Healthcare, Sydney, NSW, Australia, 21 October 2022*, pages 1–6. ACM, 2022.

[RM14] Luca Rossi and Mirco Musolesi. It's the way you check-in: identifying users in location-based social networks. In Alessandra Sala, Ashish Goel, and Krishna P. Gummadi, editors, *Proceedings of the second ACM conference on Online social networks, COSN 2014, Dublin, Ireland, October 1-2, 2014*, pages 215–226. ACM, 2014.

[RVBM09] Mariana Raykova, Binh Vo, Steven M. Bellovin, and Tal Malkin. Secure anonymous database search. In Radu Sion and Dawn Song, editors, *Proceedings of the first ACM Cloud Computing Security Workshop, CCSW 2009, Chicago, IL, USA, November 13,*

2009, pages 115–126. ACM, 2009.

[SA16] Dezan Shira and Associates. Cyber-security law of the people's republic of china, 2016.

[SAGK21] Amir Mahdi Sazdar, Nasim Alikhani, Seyed Ali Ghorashi, and Ahmad Khonsari. Privacy preserving in indoor fingerprint localization and radio map expansion. *Peer-to-Peer Netw. Appl.*, 14(1):121–134, 2021.

[Sch12] Thomas Schneider. *Engineering Secure Two-Party Computation Protocols - Design, Optimization, and Applications of Efficient Secure Function Evaluation*. Springer, 2012.

[SCYW14] Tao Shu, Yingying Chen, Jie Yang, and Albert Williams. Multilateral privacy-preserving localization in pervasive environments. In *2014 IEEE Conf. on Computer Communications, INFOCOM 2014*, pages 2319–2327, 2014.

[SGB12] Sanur Sharma, Preeti Gupta, and Vishal Bhatnagar. Anonymisation in social network: a literature survey and classification. *Int. J. Soc. Netw. Min.*, 1(1):51–66, 2012.

[SH12] Mudhakar Srivatsa and Michael Hicks. Deanonymizing mobility traces: using social network as a side-channel. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 628–637. ACM, 2012.

[SHSK15] Ebrahim M. Songhori, Siam U. Hussain, Ahmad-Reza Sadeghi, and Farinaz Koushanfar. Compacting privacy-preserving k-nearest neighbor search using logic synthesis. In *52nd Annual Design Automation Conf.*, pages 36:1–36:6, 2015.

[SKN05] Kang G. Shin, David Kotz, and Brian D. Noble, editors. *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, MobiSys 2005, Seattle, Washington, USA, June 6-8, 2005*. ACM, 2005.

[SS13] Emil Stefanov and Elaine Shi. Multi-cloud oblivious storage. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 247–258. ACM, 2013.

[STBH11] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying location privacy. In *32nd*

IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA, pages 247–262. IEEE Computer Society, 2011.

[Ste99] Jacques Stern, editor. *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*. Springer, 1999.

[STP+14] Reza Shokri, George Theodorakopoulos, Panos Papadimitratos, Ehsan Kazemi, and Jean-Pierre Hubaux. Hiding in the mobile crowd: Locationprivacy through collaboration. *IEEE Trans. Dependable Secur. Comput.*, 11(3):266–279, 2014.

[SYY+16] Nan Shen, Jun Yang, Ke Yuan, Chuan Fu, and Chunfu Jia. An efficient and privacy-preserving location sharing mechanism. *Comput. Stand. Interfaces*, 44:102–109, 2016.

[SZW+20] Hua Shen, Mingwu Zhang, Hao Wang, Fuchun Guo, and Willy Susilo. A lightweight privacy-preserving fair meeting location determination scheme. *IEEE Internet Things J.*, 7(4):3083–3093, 2020.

[TMT+15] Joaquín Torres-Sospedra, Raúl Montoliu, Sergio Trilles, Óscar Belmonte, and Joaquín Huerta. Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems. *Expert Syst. Appl.*, 42(23):9263–9278, 2015.

[TMU+14] Joaquín Torres-Sospedra, Raúl Montoliu, Adolfo Martínez Usó, Joan P. Avariento, Tomas J. Arnau, Mauri Benedito-Bordonau, and Joaquín Huerta. Ujiindoorloc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. In *2014 Intl. Conf. on Indoor Positioning and Indoor Navigation, IPIN 2014*, pages 261–270, 2014.

[USA14] FISMA USA. The federal information security modernization act of 2014 (fisma), 2014.

[WGL+18] Huandong Wang, Chen Gao, Yong Li, Gang Wang, Depeng Jin, and Jingbo Sun. De-anonymization of mobility trajectories: Dissecting the gaps between theory and practice. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society, 2018.

[WHJM18] Yufeng Wang, Minjie Huang, Qun Jin, and Jianhua Ma. DP3: A differential privacy-based privacy-preserving indoor localization mechanism. *IEEE Commun. Lett.*, 22(12):2547–2550, 2018.

[WXY+22] Zhiheng Wang, Yanyan Xu, Yuejing Yan, Yiran Zhang, Zheheng Rao, and Xue Ouyang. Privacy-preserving indoor localization based on inner product encryption in a cloud environment. *Knowl. Based Syst.*, 239:108005, 2022.

[WYH+17] Leye Wang, Dingqi Yang, Xiao Han, Tianben Wang, Daqing Zhang, and Xiaojuan Ma. Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation. In Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich, editors, *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 627–636. ACM, 2017.

[WZF+15] Rong Wang, Min Zhang, Dengguo Feng, Yanyan Fu, and Zhenyu Chen. A de-anonymization attack on geo-located data considering spatio-temporal influences. In Sihan Qing, Eiji Okamoto, Kwangjo Kim, and Dongmei Liu, editors, *Information and Communications Security - 17th International Conference, ICICS 2015, Beijing, China, December 9-11, 2015, Revised Selected Papers*, volume 9543 of *Lecture Notes in Computer Science*, pages 478–484. Springer, 2015.

[YA05] Moustafa Youssef and Ashok K. Agrawala. The horus WLAN location determination system. In Shin et al. [SKN05], pages 205–218.

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 160–164, 1982.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science*, pages 162–167, 1986.

[YJ18] Zheng Yang and Kimmo Järvinen. The death and rebirth of privacy-preserving wifi fingerprint localization with paillier encryption. In *2018 IEEE Conf. on Computer Communications, INFOCOM 2018*, pages 1223–1231, 2018.

[YJL21] Chen Yang, Zhen Jia, and Shundong Li. Privacy-preserving proximity detection framework for location-based services. In

2021 International Conference on Networking and Network Applications, NaNA 2021, Lijiang City, China, October 29 - Nov. 1, 2021, pages 99–106. IEEE, 2021.

[YVSG22] Huijie Yang, Pandi Vijayakumar, Jian Shen, and Brij B. Gupta. A location-based privacy-preserving oblivious sharing scheme for indoor navigation. *Future Gener. Comput. Syst.*, 137:42–52, 2022.

[YVV21] Vijay Kumar Yadav, Shekhar Verma, and Subramanian Venkatesan. Linkable privacy-preserving scheme for location-based services. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[YVV22] Vijay Kumar Yadav, Shekhar Verma, and S Venkatesan. Efficient and privacy-preserving location-based services over the cloud. *Cluster Computing*, pages 1–18, 2022.

[ZB11] Hui Zang and Jean Bolot. Anonymization of location data does not work: a large-scale measurement study. In Parmesh Ramanathan, Thyaga Nandagopal, and Brian Neil Levine, editors, *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking, MOBICOM 2011, Las Vegas, Nevada, USA, September 19-23, 2011*, pages 145–156. ACM, 2011.

[ZCLW18] Shaobo Zhang, Kim-Kwang Raymond Choo, Qin Liu, and Guojun Wang. Enhancing privacy through uniform grid and caching in location-based services. *Future Gener. Comput. Syst.*, 86:881–892, 2018.

[ZCN+13] Xiaoyan Zhu, Haotian Chi, Ben Niu, Weidong Zhang, Zan Li, and Hui Li. Mobicache: When k-anonymity meets cache. In *2013 IEEE Global Communications Conference, GLOBECOM 2013, Atlanta, GA, USA, December 9-13, 2013*, pages 820–825. IEEE, 2013.

[ZCZL16] Tao Zhang, Sherman S. M. Chow, Zhe Zhou, and Ming Li. Privacy-preserving wi-fi fingerprinting indoor localization. In *Advances in Information and Computer Security - 11th Intl. Workshop on Security, IWSEC 2016*, pages 215–233, 2016.

[ZHC+22] Xuejun Zhang, Fucun He, Qian Chen, Xinlong Jiang, Junda Bao, Tongwei Ren, and Xiaogang Du. A differentially private indoor localization scheme with fusion of wifi and bluetooth fingerprints in edge computing. *Neural Comput. Appl.*, 34(6):4111–

4132, 2022.

[ZHS+22] Peng Zhang, Teng Huang, Xiaoqiang Sun, Wei Zhao, Hongwei Liu, Shangqi Lai, and Joseph K Liu. Privacy-preserving and outsourced multi-party k-means clustering based on multi-key fully homomorphic encryption. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[ZRE15] Samee Zahur, Mike Rosulek, and David Evans. Two halves make a whole - reducing data transfer in garbled circuits using half gates. In *EUROCRYPT 2015 - 34th Annual Intl. Conf. on the Theory and Applications of Cryptographic Techniques, Proc., Part II*, pages 220–250, 2015.

[ZZXM09] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from GPS trajectories. In Juan Quemada, Gonzalo León, Yoëlle S. Maarek, and Wolfgang Nejdl, editors, *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 791–800. ACM, 2009.