2-2016

Efficient Location Privacy In Mobile Applications

Erald Troja
City University of New York, Graduate Center

EFFICIENT LOCATION PRIVACY IN MOBILE APPLICATIONS

by

Erald Troja

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2016

This manuscript has been read and accepted by the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

Dr. Spiridon Bakiras

Date

Chair of Examining Committee

Dr. Robert Haralick

Date

Executive Officer

Dr. Spiridon Bakiras

Dr. Bilal Khan

Dr. Abdullah Uz Tansel

Dr. Gabriel Ghinita

Supervisory Committee

The City University of New York

Abstract

EFFICIENT LOCATION PRIVACY IN MOBILE APPLICATIONS

by

Erald Troja

Adviser: Dr. Spiridon Bakiras

Location awareness is an essential part of today's mobile devices. It is a well-established technology that offers significant benefits to mobile users. While location awareness has triggered the exponential growth of mobile computing, it has also introduced new privacy threats due to frequent location disclosures. Movement patterns could be used to identify individuals and also leak sensitive information about them, such as health condition, lifestyle, political/religious affiliations, etc. In this dissertation we address location privacy in the context of mobile applications. First we look into location privacy in the context of Dynamic Spectrum Access (DSA) technology. DSA is a promising framework for mitigating the spectrum shortage caused by fixed spectrum allocation policies. In particular, DSA allows license-exempt users to access the licensed spectrum bands when not in use by their respective owners. Here, we focus on the database-driven DSA model, where mobile users issue location-based queries to a white-space database in order to identify idle channels in their area. We present a number of efficient protocols that allow users to retrieve channel availability information from the white-space database while maintaining their location secret.

In the second part of the dissertation we look into location privacy in the context of location-aware mobile advertising. Location-aware mobile advertising is expanding very rapidly and is forecast to grow much faster than any other industry in the digital era. Unfortunately, with the rise and expansion of online behavioral advertising, consumers have grown very skeptical of the vast amount of data that is extracted and mined from advertisers today. As a result, the consensus has shifted towards stricter privacy requirements. Clearly, there exists an innate conflict between privacy and advertisement, yet existing advertising practices rely heavily on non-disclosure agreements and policy enforcement rather than computational privacy guarantees. In the second half of this dissertation, we present a novel privacy-preserving location-aware mobile advertisement framework that is built with privacy in mind from the ground up. The framework consists of several methods which ease the tension that exists between privacy and advertising by guaranteeing, through cryptographic constructions, that (i) mobile users receive advertisements relative to their location and interests in a privacy-preserving manner, and (ii) the advertisement network can only compute aggregate statistics of ad impressions and click-through-rates. Through extensive experimentation, we show that our methods are efficient in terms of both computational and communication cost, especially at the client side.

ACKNOWLEDGMENTS

Over the past years I have received support and encouragement from the following people that I would like to thank. First and foremost I would like to express my heartfelt gratitude to my adviser, Dr. Spiridon Bakiras for his excellent mentoring, support and assistance. His persistence guidance and patient optimism were essential throughout my Phd journey. For that I am forever grateful.

I would also like to thank the other members of my dissertation committee, namely Dr. Bilal Khan, Dr. Abdullah Uz Tansel and Dr. Gabriel Ghinita for their comments and support especially in the later phase of my research. I would like to thank the previous CUNY Graduate Center Executive Director, Dr. Theodore Brown, and the current Executive Director, Dr. Robert Haralick for providing me with conference travel financial support.

Last but not least, I would like to thank the following members of my wonderful immediate family for their support, patience and continuous love throughout my studies. It has not been an easy process, yet my beloved wife, Alessandra, has always found a way to cheer me up and encourage me, despite carrying on a full time job and full time study schedule on her own. My daughter, Sophia, has kept my spirit alive and young with her silly ways during our daily piano practice. My parents, Petrika and Eli have given me unconditional love and support throughout my life. They continue to inspire me during hard times with their persistence and wisdom. My sister Hesiona and my two aunts, Alma and Roza have been very helpful and accommodating in so many ways. Without their implicit support I would not be able to dedicate enough time towards my studies. I love you all very much.

This research was sponsored in part by NSF CAREER Award IIS-0845262.

To God and to my beloved family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Mobile context awareness can be defined as a mobile device's ability to identify and infer various information which can be related to location, time, social status, physiological condition of the user etc. Context awareness can be leveraged by a mobile device in order to enhance some underlying computational function. For example, while scheduling a taxi pickup, a mobile device can leverage its GPS positioning system and provide the underlying application with its exact location coordinates. A mobile recommendation system can leverage location, as well as social status information, in order to recommend more relevant products to close friends and family. Mobile health and fitness applications can leverage a user's location, mobility speed and heart rate in order to map running routes, compute fat burning and other physiological functions which can reported as daily health charts.

While context awareness has played a catalytic role towards the rise in popularity of mobile computing, it has done so at the expense of user privacy. Out of various potential privacy threats, the threat towards location privacy is probably the biggest one that can arise as a result of abusive treatment and mis-management of a mobile device's context awareness. Location privacy is the fundamental block against the fast growing erosion of privacy and anonymity in a digital world. Movement patterns not only could be used to identify us as individuals, but they can be used to leak sensitive information about us such as health conditions, lifestyle, political/religious affiliations, etc.

In the first section of the introduction, we discuss location privacy in the context of Dynamic Spectrum Access (DSA) technology. DSA is a promising framework for mitigating the spectrum shortage caused by fixed spectrum allocation policies. We focus on the database-driven DSA model, where mobile users issue location-dependent queries to a white-space database in order to identify idle spectrum bands in their area. In the second section of the introduction we discuss location privacy in the context of location-aware mobile advertising. Mobile users have grown very skeptic of the ever growing amount of data that is extracted and mined from advertisers today, and as a result have shifted their focus towards stricter privacy requirements. However, stricter privacy requirements enforced via policy regulations are counterintuitive to a healthy mobile advertising cycle. As such we focus on a mobile advertising model which location privacy is enabled from the ground up through cryptographic constructions.

1.1 Location Privacy in Dynamic Spectrum Access

The allocation of radio spectrum for mobile wireless networking is governed by federal agencies via a fixed (static) spectrum sharing strategy. However, with the ever growing need for mobile wireless services and applications, the static sharing method has led to the depletion of the available spectrum [65]. Furthermore, the actual usage of pre-assigned spectrum bands has been measured to have a very low average utilization. For example, in the US, the Federal Communications Commission (FCC) has reported that many spectrum bands allocated via static assignment policies have been used only in bounded geographical areas and over very limited periods of time. Such utilization has been measured to be between 15% and 85% [9].

Currently, there is wide consensus that the static method of spectrum allocation has major drawbacks. As a result, the need for opportunistic and dynamic spectrum access technologies has risen sharply. A flexible and dynamic spectrum access strategy is necessary, in order to eliminate the underutilization and spectrum depletion effects of the current static allocation scheme. The FCC has stated that no other technology "holds greater potential for literally transforming the use of spectrum in the years to come than the development of software-defined and cognitive/smart radios" [24].

To this end, DSA allows users to access licensed spectrum bands when not in use by their respective owners. DSA is built on top of Cognitive Radio (CR), an intelligent wireless communications system that is aware of its spectral environment [63]. A CR node must be able to dynamically adapt to the environmental spectral changes in order to abide by the spectral etiquette set forth by the FCC. One of the most important functions that a CR node must perform is the identification of unoccupied spectrum opportunities (SOPs). SOPs are space, time, and frequency dependent blocks, during which the license-exempt can utilize the registered owner's spectrum in a DSA manner. Prior to May 2012, SOP discovery was mainly done through distributed and cooperative sensing. In such an approach, CR nodes rely on sheer power detection methods, and coordinate in order to identify spectrum activity and locate available SOPs [41, 91, 95, 96].

Alternatively, in database-driven dynamic spectrum access, a CR node understands its spectral surroundings in a three-step process. A node attempting to analyze its spectral surrounding would first learn its geographic location through a GPS device. Subsequently, it would contact a centralized white-space database (WSDB) and issue its GPS coordinates as part of the query. Finally, it would download the centrally fused repository report containing the available spectrum at that location. The compilation and fusion of the spectrum reports is done by specialized entities, called Spectrum Database Operators (SDOs), by applying appropriate propagation modeling and interference avoidance algorithms for a given geographic location.

FCC's May 2012 ruling [21] obsoletes the distributed and cooperative sensing method for the white-space TV bands. The ruling requires that all CR nodes operating in the white-space TV bands utilize the centralized white-space database (WSDB) spectrum lookup method. In order to allow mobile Television Band Devices (TVBDs) to learn their spectral surroundings, the FCC has designated 10 WSDB providers, out of which only Google, Spectrum Bridge, and Telcordia Technologies have been approved for operation [23].

Nevertheless, the database-driven DSA approach is prone to severe location privacy leakage. According to FCC specifications [22], a mobile Television Band Device (TVBD) must issue a new query whenever it moves further than 100m from its previous location. Since the GPS coordinates must be part of every query, a WSDB operator could easily build a detailed history of the mobile TVBD's trajectories, which could reveal sensitive information about the underlying user (such as health condition, habits, etc.).

As an example, Fig. 1.1 shows a mobile TVBD's trajectory that is formed by latitude/-longitude data points taken at consecutive time intervals, near Tsinghua University. Given the starting point of the trajectory, the WSDB server can identify (to a certain extent) the user associated with this trajectory (e.g., it may correspond to a home address). In addition, given the end point of the trajectory, the WSDB server can infer (with a certain probability) that the aforementioned TVBD user is affiliated with Tsinghua University.

To this end, Gao et al. [27] introduce a scheme that leverages a private information retrieval (PIR) protocol to query the WSDB in a privacy-preserving manner. A PIR protocol allows any user to retrieve a record from a database server, while maintaining the identity of the record secret from the server. Therefore, Gao et al. partitions the space with a fixed $n \times n$ grid and requires users to download the location-dependent (based on the cell where they are located) channel information, through the PIR protocol. This is the only protocol so far in the literature dealing with location privacy in database-driven DSA but, unfortunately, it suffers from several drawbacks.

First, Gao et al. utilizes the PIR scheme of Trostle and Parrish [86] whose communication cost (for a single query) is significant. More precisely the scheme incurs a communication cost

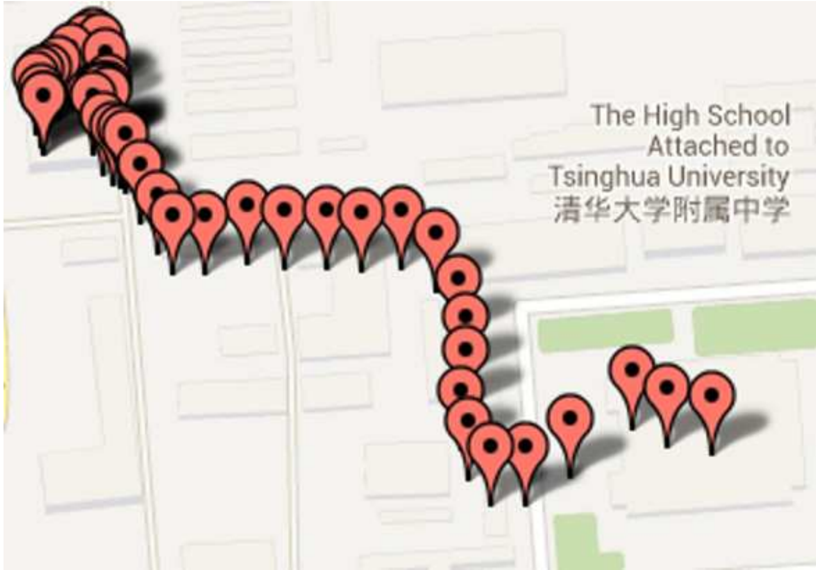


Figure 1.1: Mobile user geo-located near Tsinghua University (from Microsoft's GeoLife trajectory dataset).

of $(2n+3) \cdot \log\ p$ bits, where p is a 2048-bit modulus. For instance, if $n = 5000$, the amount of data exchanged to retrieve the bitmap of a single cell is 2.5 MB. For highly mobile clients, the cost of this approach can exceed the cost of downloading the entire database. Second, most PIR protocols typically return multiple records per query that, in the case of mobile users, could be used to answer future queries. However, the authors modify [86] so that the PIR reply contains channel availability information for a *single* cell (as opposed to n in the original protocol). Finally, they view each query as an independent event, without taking into account user mobility. As a result, when a user is constantly moving, the communication cost can surpass the cost of downloading the entire database.

Our contribution

In the first part of the contribution towards location privacy in database-driven dynamic spectrum access, we first argue that dynamic spectrum access will most likely be utilized in areas with poor/intermittent cellular connectivity. As such, the underlying query processing protocol should be communication efficient. Therefore, unlike [27], our methods leverage the PIR scheme of Gentry and Ramzan [30], which is the most communication efficient protocol to date. Furthermore, to address user mobility, we index the WSDB based on the Hilbert space filling curve (HSFC) [50]. In this way, neighboring cells are typically stored in consecutive locations on the white-space database. Finally, to allow for the retrieval of multiple cells with a single PIR query, we split the WSDB into multiple, disjoint segments. As such, a PIR query is processed independently on each segment, and the user retrieves channel availability information from a large number of consecutive cells IDs. Due to the properties of the underlying HSFC, these cells will be spatially close (with a very high probability), and could reduce the number of PIR queries in the near future.

We consider two distinct cases in our work: (i) the user's trajectory is known a priori and (ii) the user's trajectory is generated on-the-fly. For the latter case, we propose a trajectory prediction method, based on a simple linear regression model of the recently traveled coordinates. The predicted values are then used to retrieve the corresponding cells from the WSDB and, thus, reduce further the number of future PIR queries. In the case of the a priori trajectory knowledge, our approach enables mobile users to simulate their routes and invoke the lowest number of PIR queries. We tested our methods on two real life datasets, namely Microsoft's T-Drive dataset [93] and Microsoft's GeoLife GPS dataset [97]. The experimental results show that, compared to the protocol of Gao et al. [27], our methods reduce the query response time at the mobile clients by at least a factor of 30.

Nonetheless, typical PIR protocols offer a trade-off between computational and communication complexity. Computational complexity has an adverse impact mostly at the server side, whereas communication complexity affects the end-user as well (especially in the case of wireless devices). For instance, the scheme by Trostle and Parrish [86] that is applied in previous work [27] is computationally efficient, but its communication cost is equal to a large percentage of the database size. On the other extreme, Gentry and Ramzan's protocol [30] which we use in the initial contribution of this dissertation is considered to attain the best communication complexity, but incurs a high computational cost due to its heavy use of cryptographic operations [68].

Therefore, we argue that any location privacy method for the database-driven DSA model is bounded by the limitations of the underlying PIR protocol. As such, it is desirable to identify new mechanisms for users to acquire the necessary spectral knowledge. Our intuition is that, in a white-space TV band network, mobile TVBD users will gradually develop a trajectory-specific spectrum knowledge *cache*, through a series of PIR requests. In the extreme case, some users might opt to download the entire WSDB (trivial PIR case) before initiating their travel[1]. Therefore, as our second contribution towards location privacy in database-driven dynamic spectrum access, we propose that mobile users that are within communication range interact in a peer-to-peer (P2P) manner, in order to exchange their cached spectrum knowledge for the surrounding area.

However, a user's spectrum knowledge cache is a summary of his/her recent trajectory, and some users may be unwilling to share that information due to privacy concerns. To this end, we leverage the *anonymous veto network* (AV-net) protocol of Hao and Zieliński [39] that anonymizes the exchange of information among a group of users. Our experimental results with Microsoft's GeoLife trajectory dataset [97] show that our methods reduce the number of PIR queries by 50% to 60%, while incurring low computational and communication costs for the mobile clients.

1.2 Location Privacy in Mobile Advertising

In the second part of the dissertation we shift our focus towards the preservation of location privacy in location-aware mobile advertising. According to Gartner [28], a leading information technology research and advisory firm, mobile computing is forecast to continue its

[1] However, due to its overwhelming communication cost, the trivial PIR case may be infeasible for most users.

rapid expansion in the foreseeable future. Given the amount of time consumers spend on their smartphones and mobile devices, the market for mobile advertising will continue to grow steadily, soaring to an 18 billion dollars/year industry. In fact, that figure will surpass traditional newspaper advertising budgets for the first time ever [89]. The success of mobile advertising can be attributed in part to the mobile devices' capability of collecting various interesting measurements, such as movement speed and direction, elevation, latitude and longitude, etc. This is the foundation of many popular applications today, including Foursquare, GasBuddy, Waze, Trigger etc.

Unfortunately, the rise of popular smartphone applications has led to an exponential growth in privacy concerns, mainly due to the constant consumer tracking and profiling that stems from abusive data capturing and sharing strategies. According to a recent study [87], 66% of location-aware applications have privacy policies that actually amount to very little when it comes to protecting the privacy of the data that is collected and shared by the application. A classic case study is that of the Google engineer who violated the company's strict privacy policy rules, by breaking into the Gmail and Voice accounts of Google users [15]. To this end, various policy initiatives have been implemented around the world in order to protect user privacy.

In the U.S., privacy initiatives such as the Federal Trade Commission's (FTC) *DoNot-TrackMe* option give consumers a choice on whether to participate in online behavioral advertisements. In Europe, initiatives such as the *Right To Be Forgotten* attempt to partially address the privacy concerns of European consumers. Nevertheless, such initiatives, which are considered as band-aid solutions at best, apply only to website operators, and not necessarily to mobile applications running on a consumer's smartphone. Therefore, location-aware mobile advertisement applications can easily bypass such policy enforcements. Furthermore, strict policies hamper the success of mobile advertising, since they offer either an all-in or an all-out choice for the consumers. Often, free versions of popular mobile applications are fueled by mobile advertising revenue, and such policy initiatives disrupt the targeted mobile advertising cycle, thus negatively impacting app developers.

To eliminate the privacy concerns of online targeted advertisement, a privacy-preserving *ad network* should address two important issues, namely ad delivery and statistics collection. First, ad delivery should guarantee that the network is oblivious to the content sent to the clients. Second, statistics collection should keep track of the aggregate number of times that a certain ad is displayed to the users (known as ad impression or click-through-rate) without knowledge of individual statistics. These aggregate measurements are essential, because they form the basis on which the ad network bills the advertisers. Although several privacy-preserving ad networks exist in the literature, some do not support location-aware ads [48, 79], others do not encrypt the ads sent to the clients [38, 40, 79], a few employ a trusted third-party in their architecture [37, 79], others rely on non-colluding servers [40, 48], and some require specialized network infrastructures [38].

Our contribution

In the second part the dissertation, we introduce a novel privacy-preserving location-aware mobile advertisement framework that is built with privacy in mind from the ground up. Our methods ease the tension that exists between privacy and advertising by guaranteeing, through cryptographic constructions, that (i) mobile users receive advertisements relative to their location and interests encrypted with their own public keys, and (ii) the ad network can only compute aggregate statistics of ad impressions and click-through-rates (CTRs). Unlike previous work, we do not employ trusted third-parties and all our protocols are secure against collusions. To the best of our knowledge, this is the first privacy-preserving location-aware ad network in the literature with such properties.

Our basic ad delivery method leverages a simplified version of the private stream searching protocol by Ostrovsky and Skeith III [66]. Specifically, we partition space into a regular grid, and allow clients to privately retrieve the ads from the grid cell that they currently reside in. This is done by sending to the ad network's server one Paillier [67] ciphertext for each grid cell. All ciphertexts contain encryptions of 0, except for the one corresponding to the client's cell that contains an encryption of 1. With these ciphertexts, the server prepares an encrypted ad buffer that is then decrypted by the client with his private key. Next, we identify a performance bottleneck at the client side when the number of cells is large, which leads to significant computational and communication costs. To this end, we introduce an improved protocol, based on the "somewhat" homomorphic cryptosystem of Boneh, Goh, and Nissim (BGN) [5]. The properties of the BGN cryptosystem allow us to identify the cell of interest through its row/column id and, therefore, does not require a unique ciphertext for each grid cell.

Our final contribution includes a privacy-preserving aggregation protocol that collects ad impression/CTR statistics at the ad network server, without leaking any information regarding individual customers. It is based on a distributed version of the ElGamal cryptosystem [17] and has several desirable properties. First, it is very efficient for dynamic membership groups, i.e., when new customers join or existing customers leave the system. Furthermore, it is resistant against collusions among the clients, and is computationally efficient. Through extensive experimentation, we show that our protocols are efficient in terms of both computational and communication cost, especially at the client side.

1.3 Dissertation Outline

This dissertation is organized as follows. Chapters 2 and 3 present the details of our contribution on location privacy as it relates to the database-driven Dynamic Spectrum Access framework. Chapter 4 introduces our work in location privacy as it relates to mobile advertisement. Chapter 5 provides conclusions and lays out our future research plans.

Accepted Publications Contained in this Dissertation

This thesis contains work which has been already published and/or is submitted for publication as follows:

- Erald Troja and Spiridon Bakiras. Efficient location privacy for moving clients in database-driven dynamic spectrum access. In *Proceedings of the 24th International Conference on Computer Communications and Networks*. IEEE, 2015 [82]. This publication forms the basis for the protocol descriptions and performance measurements mentioned in Chapter 2.

- Erald Troja and Spiridon Bakiras. Leveraging P2P interactions for efficient location privacy in database-driven dynamic spectrum access. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 453–456. ACM, 2014 [81].

- Erald Troja and Spiridon Bakiras. Leveraging P2P interactions for efficient location privacy in database-driven dynamic spectrum access. *International Journal of Network Security*, 17(5):569–579, 2015 [83]. These publications form the basis for the protocol descriptions and performance measurements mentioned in Chapter 3.

- Erald Troja and Spiridon Bakiras. Privacy-preserving location-aware mobile advertisement. *In-submission* [80]. This publication forms the basis for the protocol descriptions and performance measurements mentioned in Chapter 4.

CHAPTER 2

EFFICIENT LOCATION PRIVACY FOR MOVING CLIENTS IN DATABASE-DRIVEN DSA

2.1 Background

Most existing approaches for location privacy rely on the notion of k-anonymity [77] or l-diversity [61]. In location-based services, a spatial query is said to be k-anonymous, if it is indistinguishable from at least $k-1$ other queries originating from the same region. This region is called a spatial cloaking region (SCR), and encloses the querying user as well as at least $k-1$ other users. To compute the SCR, existing k-anonymity algorithms typically extend the SCR around the query point until it encloses $k-1$ other users [36].

l-diversity based methods [61], on the other hand, extend the SCR until $l-1$ different locations are included. Although k-anonymity and l-diversity provide some degree of location privacy, they may still leak semantic location information. For example, if the SCR only contains casinos, the server can infer that the mobile user is interested in gambling. To this end, the work of Lee et al. [54] attempts to provide location privacy using location semantics.

The k-anonymity and l-diversity based approaches, as well as collaborative location privacy protection methods [25], often rely on third party trusted anonymizers, which is not always a viable solution. On the other hand, Ghinita et al. [31] propose the first privacy-preserving protocol (for nearest neighbor queries) that does not require a trusted third party.

Instead, their method achieves perfect location privacy via the cryptographic primitive of private information retrieval [53].

Other protocols on location privacy revolve around the notion of data perturbation, location hiding, and the introduction of data point dummies. Meyerowitz et al. [62] introduce a data perturbation technique to protect personal location data against untrusted location based service (LBS) servers. In their work, they develop CacheCloak, a protocol that enables real time anonymization of location data. CacheCloak relies on a trusted anonymizing server to generate mobility prediction from historical data, and then submit intersecting predicted paths simultaneously to the LBS. Reliance on a trusted server is a very expensive and strong assumption that we would like to avoid in our proposed methods. Also, the intuition behind CacheCloak is to obscure the user's path by surrounding parts of it with other user's paths, effectively creating a k-anonymous region.

Huang et al. [43] study the problem of location privacy preservation with respect to an LBS that threatens a user's location privacy by tracking transmitting frames. The authors argue that correlation attacks between a node's old and new address are not sufficient. They suggest the concept of a *silent period*, defined as the transition period between the use of new and old pseudonyms, during which a node is not allowed to disclose neither the old nor the new address.

Furthermore, Huang et al. [44] extend their previous work [43] and study the problem of location privacy with respect to a user's communication with network access points. They mainly focus on the issue of how location privacy enhancements affect the perceived Quality of Service (QoS). The authors propose a silent cascade method to enhance a user's location privacy by trading end-to-end delay for anonymity. They abstract silent cascade as a mix-network model and evaluate its performance. In our setting, however, we are concerned with the effectiveness (computation and communication cost) of the location privacy-preserving protocol itself.

Kido et al. [52] suggest a location privacy-preserving method that uses the notion of *dummy* data (false positives), in order to hide the user's true location from the LBS. The authors argue that, after sending their GPS coordinates to the LBS, users can not delete or modify their disclosed location. In other words, users cannot prevent the service providers from analyzing motion patterns using stored location data. In their proposed method, users send their true location data along with several false ones (dummies) to the service provider, who subsequently creates a reply message for each received data point. Users then simply extract the correct information from the reply messages. However, it is clear that this scheme is essentially a k-anonymity based approach.

Similarly to Kido et al. [52], Lu et al. [60] introduce PAD, a method that injects dummy locations in the query, which are generated according to either a virtual grid or a circle. The virtual grid or circle cover the user's actual location, and their spatial extents are controlled by appropriate generating algorithms. However, PAD relies on a server-side front-end, in order to be integrated into existing client/server mobile service systems. Even though PAD takes into account the number of location points in the query, as well as the area of the region covered by those points, it can be effectively reduced to a pure k-anonymity based technique.

Other techniques such as routing anonymization and privacy-preserving wireless broadcast networks have been suggested. The authors in [3] suggest wireless anonymous routing (WAR) as the main approach of achieving anonymity in a wireless broadcast network. Ref. [98] and [55] propose lightweight ad hoc routing protocols in order to preserve location privacy of the mobile nodes. Lastly, the authors in [45] provide evidence that such anonymization and location privacy-preservation techniques can be applied even in radio frequency identification networks (RFID). Such techniques are orthogonal to our proposed methods and can be applied in an optional and complementary fashion in order to provide local network addressing anonymity as well as geo-location privacy.

Until recently, location privacy work in the dynamic spectrum access domain has mainly focused on the collaborative spectrum sensing model. In particular, most existing solutions attempt to protect the location privacy of mobile users that submit sensing reports to a fusion center [42, 57, 76]. The collaborative sensing and reporting approach was embraced as a superior method compared to the centralized database approach. This is no longer the case, though, at least in the white-space TV band realm.

Due to the recency of the FCC's ruling (May 2012), location privacy research in database-driven DSA networks is still in its infancy. The state-of-the-art protocol is due to Gao et al. [27], which builds upon a modified version of Trostle and Parrish's PIR scheme [86]. They assume a fixed grid of $n \times n$ cells, where each cell contains a bitmap that represents the channel availability information (typically 32 bits). The authors modify [86] so that the PIR reply contains channel availability only for a *single* cell. Furthermore, each query is seen independently, without any regard to user mobility. Their scheme incurs a high communication cost of $(2n + 3) \cdot \log p$ bits, where p is a 2048-bit modulus. For example, if $n = 5000$, the amount of data transmitted to retrieve the bitmap of a single cell is 2.5 MB, which is approximately 2.6% of the whole database size. For highly mobile clients, the cost of this approach can exceed the cost of downloading the entire database.

In the following section we give a brief description of PIR protocols and Hilbert space filling curves. Section 2.2.1 introduces the concept of private information retrieval and Section 2.3 describes the threat model of our methods. Section 2.2.2 presents the Hilbert space filling curve algorithm.

2.2 Preliminaries

2.2.1 Private Information Retrieval

PIR protocols allow a user to obtain information from a database server, in a manner that prevents the database from knowing which data was retrieved. Typically, the server holds a database of N records and the user wants to retrieve the i-th record, such that i remains unknown to the database. The trivial PIR case consists of downloading the entire database, which clearly preserves privacy but has an unrealistic communication cost. Therefore, the objective of a PIR protocol, as applied to mobile applications, is to reduce the communication cost.

Information theoretic PIR protocols [13] are secure against computationally unbounded adversaries. However, they require that the database be replicated into multiple non-colluding servers. This non-collusion assumption is not realistic in typical applications, so information theoretic protocols are not utilized in practice. On the other hand, *computational* PIR (CPIR) protocols base their security on well-known cryptographic problems that are hard to solve (such as discrete logarithm or factorization). As such, their security is established against computationally bounded adversaries. Kushilevitz and Ostrovsky [53] introduced the first CPIR protocol for a single database, whose security is based on the quadratic residuosity assumption. The communication complexity of [53] is $O(n^{\epsilon})$. Further work [7, 59] demonstrates CPIR schemes with polylogarithmic communication complexity.

In this work, we leverage the protocol of Gentry and Ramzan [30], because it is the most communication efficient PIR protocol to date. For a particular instantiation, it exhibits a *constant* communication cost that is independent of the database size (it typically involves the exchange of three 128-byte numbers). The security of the protocol is based on "ϕ-hiding" assumption. and its functionality is summarized in [30].

Setup. We assume a database of N records, where each record is of size ℓ bits. During a preprocessing phase, the server associates every record j with a prime power $\pi_j = p_j^{c_j}$, where p_j is a small prime and c_j is the smallest integer, such that $\log \pi_j > \ell$. All these values are public knowledge. Next, using the Chinese Remainder Theorem (CRT), the server expresses the entire database as an integer e, which is the solution to the congruences $e \equiv B_j \pmod{\pi_j}$, for all $j \in \{1, 2, \ldots, N\}$. ($B_j$ is the binary representation of record j.) Client queries are processed on the transformed database e.

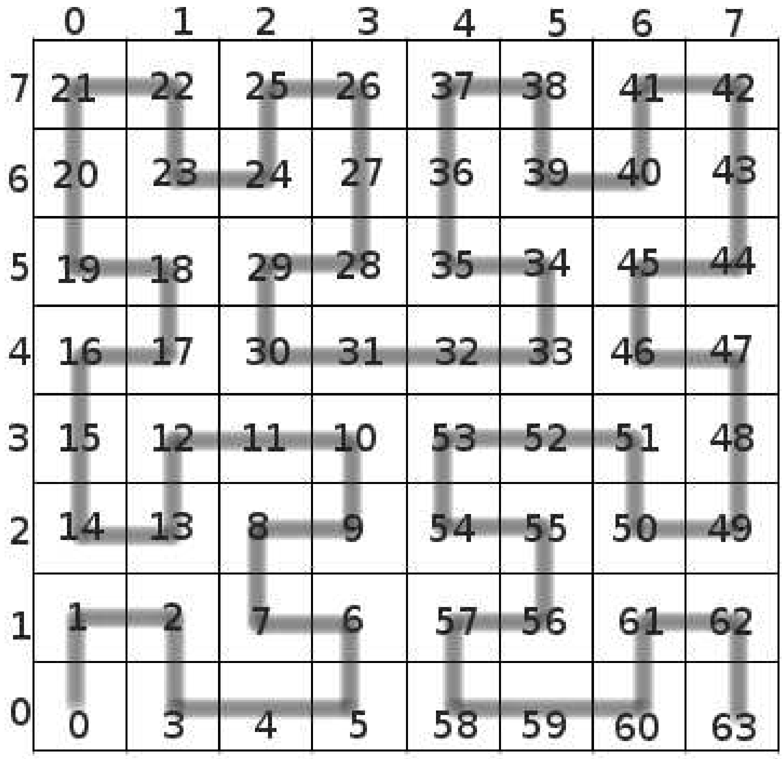
Query Generation. Groth et al. [35] show that Gentry and Ramzan's protocol can be used to retrieve multiple records with a single query. Let $i_1, i_2, \ldots, i_k$ be the indexes of the records to be retrieved. The client computes $\pi = \prod_{j=1}^{k} \pi_{i_j}$ and chooses two large prime numbers p and q, such that $p = 2\pi r + 1$ and $q = 2st + 1$, where r, s, and t are large random integers. The client sets $m = pq$ and proceeds to select a random element $g \in \mathbb{Z}_m^*$ with order πv, where $\gcd(\pi, v) = 1$. Finally, the client sends (g, m) to the server. For security reasons, it should hold that $\log m > \max(1024, 4 \log \pi)$.

Database Response. The server computes $c = g^e \mod m$ and sends the result back to the client.

Result Retrieval. To reconstruct the records, the client computes, for each i_j, $c^{\pi v/\pi_{i_j}} \mod m$, which should be equal to $(g^{\pi v/\pi_{i_j}})^{B_{i_j}} \mod m$. Therefore, the client can retrieve record B_{i_j}, using the Pohlig-Hellman algorithm for discrete logarithms [72].

2.2.2 Hilbert Space Filling Curve

The Hilbert space filling curve [50] is a continuous fractal that maps coordinates from 2-D to 1-D. If (x, y) are the coordinates of a point within the unit square and d is the distance along the curve when it reaches that point, then points with nearby d values will also be spatially close. As an example, Fig. 2.1 shows a HSFC of level $l = 3$, containing $4^l = 64$ cells. Each of the cells is identified by its (x, y) coordinates, starting with $(0, 0)$ on the lower left hand corner and ending with ($x = 2^l - 1, y = 2^l - 1$) on the right upper hand corner. The values shown in the individual cells correspond to their Hilbert IDs (HIDs), i.e., their specific order within that mapping. Note that, the Hilbert function could be initialized on any of the four corners $(0, 0)$, $(0, 7)$, $(7, 7)$, and $(7, 0)$, without affecting the 2-D to 1-D mapping. The mapping of points to their Hilbert IDs might change, but it would still preserve locality.



	0	1	2	3	4	5	6	7
7	21	22	25	26	37	38	41	42
6	20	23	24	27	36	39	40	43
5	19	18	29	28	35	34	45	44
4	16	17	30	31	32	33	46	47
3	15	12	11	10	53	52	51	48
2	14	13	8	9	54	55	50	49
1	1	2	7	6	57	56	61	62
0	0	3	4	5	58	59	60	63

Figure 2.1: A level 3 Hilbert space filling curve.

2.3 Threat Model and Security

In this work we are concerned with privacy against the WSDB operator (adversary). We assume that the adversary's goal is to derive any relevant information regarding the location of any user that has sent a query to the database. We also assume that the adversary runs in polynomial time and follows the *honest-but-curious* adversarial model, i.e., it follows the protocol correctly but tries to gain an advantage by examining the communication transcript. Note that our methods inherit the security of the underlying PIR protocol, since the only interaction between the WSDB operator and the users is through a series of PIR invocations.

2.4 Details

In this section, we present the details of our methods. Section 2.4.1 describes the system architecture, and Section 2.4.2 introduces our basic approach. Section 2.4.3 presents an enhanced method that retrieves multiple cells from the area surrounding the query point. Section 2.4.4 introduces our best algorithm that incorporates trajectory prediction, and Section 2.4.5 describes the case where there is full a priori trajectory knowledge.

2.4.1 System Architecture

Similar to previous work [27], we assume a fixed grid of $n \times n$ cells. According to the FCC specifications [22], each cell is 100m×100m in size, and users must query the WSDB whenever they move into a cell with no prior spectrum availability knowledge. The dimensions of the grid (i.e., n) can be made arbitrarily large, which has a direct effect on the database size. Mobile TVBDs are allowed to communicate only in the frequency ranges 512-608 MHz (TV channels 21-36) and 614-698 MHz (TV channels 38-51), i.e., there are a total of 31 possible white-space TV band channels that can be accessed in a DSA manner. Therefore, we represent the daily channel availability as 32 bits (per cell), where bit 0 represents a busy channel and bit 1 represents an idle channel. As an example, when $n = 5000$, the WSDB is 100 MB in size, which would take approximately 35 mins to download on a 3G network[78].

2.4.2 Single Row Retrieval

Papadopoulos et al. [68] conducted a in-depth study of the PIR protocol by Gentry and Ramzan [30] that we employ in our methods. As they point out, due to the security constraints of the algorithm, the optimal strategy in terms of communication and computational cost is to set the size of each record to 32 bytes. Therefore, based on our system settings, each record can store channel availability information from 8 distinct cells. A straightforward implementation would then be to (i) sort the cells based on their unique Hilbert IDs, and (ii) create a single database (DB) with $N = \lceil n^2/8 \rceil$ records, such that record 0 stores cells 0–7, record 1 stores cells 8–15, etc. (Table 2.1 summarizes the symbols used in the remainder of this paper.) In the toy example of Fig. 2.1, we would have a database of $N = 8$ records, and a user located inside cell 30 would retrieve the record containing cells 24–31. Observe that, due to the properties of the HFSC, all the retrieved cells are spatially close and could be useful in subsequent queries.

Table 2.1: Summary of Symbols

Symbol	**Description**
n	Number of rows/columns in the grid
k	Number of DB segments
N	Number of records in each DB segment ($N = \lceil n^2/8k \rceil$)
u	Number of records retrieved from each DB segment
$\log m$	Size of PIR request/reply (Gentry-Ramzan)
R	Number of rings to explore in the surrounding area

Nevertheless, the single DB approach would not work well in practice. First, it is beneficial for a client to retrieve a large number of cells that are in proximity to his current location, in order to reduce the number of future PIR queries. Second, Gentry and Ramzan's protocol is computationally expensive (at the server side), due to its heavy use of cryptographic operations. As such, we would like to parallelize its operation, to the extent possible, by utilizing large CPU clusters that are typical in most cloud computing platforms.

The obvious solution to both limitations is to partition the database into k distinct segments. By doing so, we can employ k CPUs to process each segment in parallel, thus reducing the computational time by a factor of k. The price we have to pay is an increase in the communication cost, since the client receives k PIR replies instead of one. Specifically, the communication cost is equal to $(2 + k) \cdot \log m$, where m is an RSA modulus. Table 2.2 shows a sample DB segmentation (for $k = 4$) for a level 4 HFSC, containing 256 cells. The segments are constructed by assigning the original records to each segment in a round-robin manner.

Table 2.2: Sample DB segmentation with 4 segments

DB segment 0	DB segment 1	DB segment 2	DB segment 3
0–7	8–15	16–23	24–31
32–39	40–47	48–55	56–63
64–71	72–79	80–87	88–95
96–103	104–111	112–119	120–127
128–135	136–143	144–151	152–159
160–167	168–175	176–183	184–191
192–199	200–207	208–215	216–223
224–231	232–239	240–247	248–255

Algorithm 1 lists the detailed algorithm for the single row retrieval method. During query processing, the client first identifies the row r that contains his current cell's HID ($r = HID/8k$). He then constructs the corresponding PIR query that is processed on all k DB segments, in parallel. In the example of Table 2.2, if the client is located in cell 180, he will retrieve all cells in row $r = 180/32 = 5$, i.e., all cells within the range 160–191. The results are stored in the client's cache and may be utilized when the client moves into a new cell.

Algorithm 1 Single Row Retrieval

```
procedure SINGLE-ROW-RETRIEVAL($HID, k$)

    if ($HID \notin cache$) then
        $r \leftarrow HID/8k$;
        $cells[\,] \leftarrow PIR(r)$;
        $cache \leftarrow cells[\,]$;
    end if

end procedure
```

2.4.3 Exploring the Surrounding Area

When a mobile user's trajectory is generated on-the-fly, i.e., without any prior planning, retrieving a single row per PIR query is not the optimal strategy. Consider, for example, a user that issues a PIR query from the cell marked with a white dot in Fig. 2.2. The numbered boxes in this figure indicate the cells that comprise the corresponding database rows. According to that figure, the user first retrieves row 6 and then moves to the next location that is part of row 8 (the black dots show the remaining trajectory points). He now has to send a new query to the WSDB and all the information contained in row 6 is rendered useless.

A second drawback of the single row retrieval approach, is the structure of the Hilbert curve itself. As evident in Fig. 2.1, a cell's nearest neighbors are not always mapped on consecutive Hilbert IDs. For instance, cells 5 and 58 are direct neighbors on the grid, but their Hilbert IDs are very far apart. These inconsistencies are common in all space filling curves, and are more severe on higher level curves (which is typically the case in real life applications).

To address these shortcomings, we take advantage of Gentry and Ramzan's multi-record retrieval feature, as described by Groth et al. [35]. Specifically, given a database segment containing N 32-byte records, we partition the segment into u sub-segments, each storing

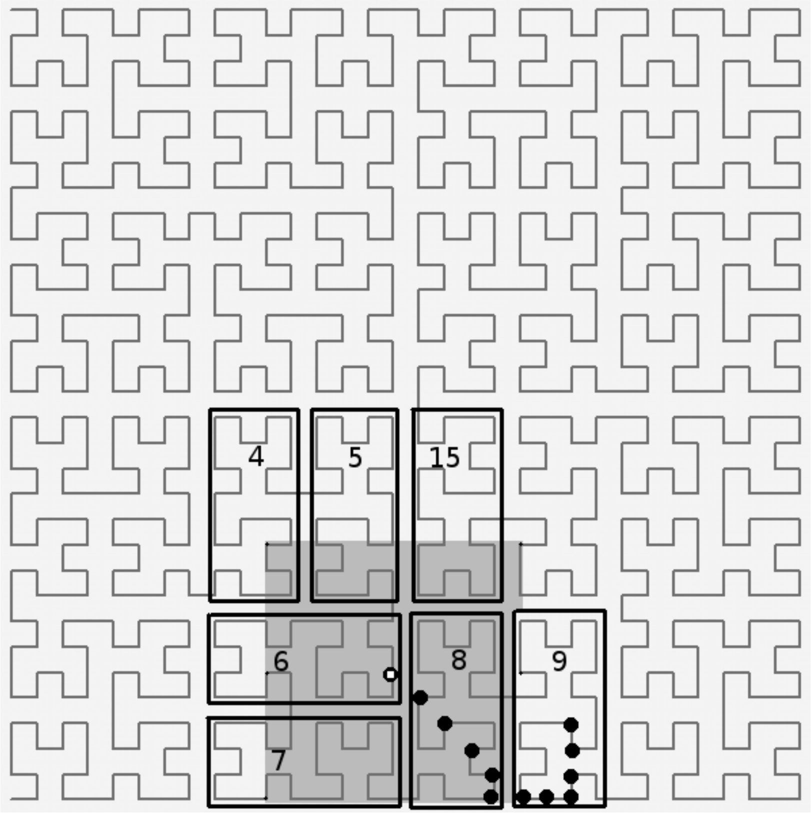


Figure 2.2: Exploring the surrounding area with 2 and 4 sub-segments.

N $(32/u)$-byte records. By doing so, it is possible to retrieve u records from each sub-segment, while keeping the computational cost unchanged. Therefore, by sacrificing some communication cost, we can retrieve more relevant results with a single query. Note that, the communication cost in the multi-record retrieval scheme is $(2 + k \cdot u) \cdot \log m$.

As a first step towards improving our basic scheme, we require the user to explore the area surrounding his current location, and retrieve the database rows that maximize the coverage of that area. The intuition is that, if the user has no prior knowledge of his trajectory, we should anticipate his movement towards any possible direction. Algorithm 2 illustrates the functionality of this approach. We define as R the number of rings surrounding the user's current cell that we want to explore.

The algorithm maintains an array *rows*, which stores the row numbers that should be

Algorithm 2 Surrounding Area

```
 1: procedure SURROUNDING-AREA($HID, k, u, R$)
 2:
 3:     $count[N] \leftarrow \{0\}$;
 4:
 5:     if ($HID \notin cache$) then
 6:         $r \leftarrow HID/8k$;
 7:         insert $r$ into $rows[\,]$;
 8:         for each cell $i$ in the area defined by $R$ do
 9:             $r' \leftarrow hid(i)/8k$;
10:             if ($r' \neq r$ and $hid(i) \notin cache$) then
11:                 $count[r']++$;
12:             end if
13:         end for
14:         find the top $(u-1)$ values in array $count$;
15:         insert their indexes into $rows[\,]$;
16:         $cells[\,] \leftarrow PIR(rows[\,])$;
17:         $cache \leftarrow cells[\,]$;
18:     end if
19:
20: end procedure
```

retrieved from the database. The first row is always the one containing the user's current cell (lines 6–7). Next, the algorithm iterates over all cells within the area defined by R, and counts how many times the underlying row numbers appear in the result (lines 8–13). Finally, it selects the $(u-1)$ most frequent row numbers and adds them into *rows* (lines 14–15). The cells from all u rows are then retrieved via the PIR query and are eventually cached at the client. In the example of Fig. 2.2, when $u = 2$ we retrieve rows 6 and 8. On the other hand, when $u = 4$ we retrieve rows 6, 8, 7, and 15.

2.4.4 Trajectory Prediction

Even if a mobile user is unaware of his exact trajectory, he is very likely to occasionally follow a specific direction (e.g., south-east) for a sufficiently large period of time. Therefore, in our next method, we explore the feasibility of employing a trajectory prediction algorithm, in order to maximize the amount of useful information retrieved from a PIR query. To this end, we assume that the client maintains a cache v of his most recent GPS measurements that are taken at regular time intervals.

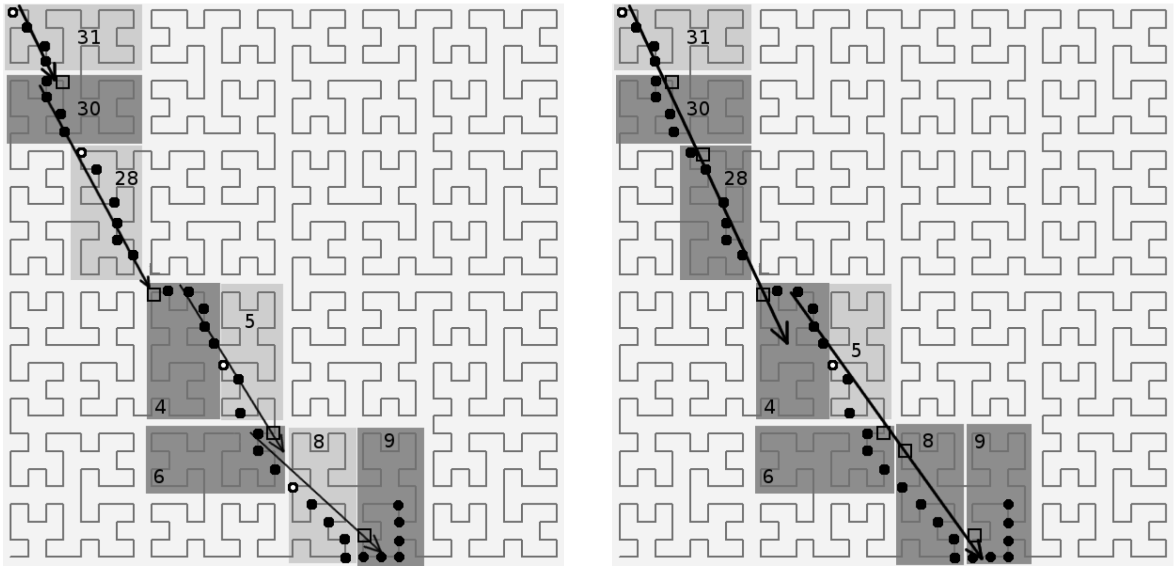
Algorithm 3 shows the detailed steps of this approach. As in our previous method, we retrieve a total of u rows, where the first row is always the one containing the user's current cell. Next, the client applies a simple linear regression (SLR) model on the vector v of GPS measurements, and computes a straight line l that predicts the following trajectory points (line 6). This line is then extended forward, until it encounters $(u-1)$ additional cells whose underlying rows are not present in the cache. The row numbers of these cells are also added to the PIR query (lines 7–11).

Algorithm 3 Trajectory Prediction

```
 1: procedure TRAJECTORY-PREDICTION($HID, k, u, v$)
 2:
 3:     if ($HID \notin cache$) then
 4:         $r \leftarrow HID/8k$;
 5:         insert $r$ into $rows[\,]$;
 6:         $l \leftarrow SLR(v)$;
 7:         for $i = 1$ to $(u-1)$ do
 8:             extend $l$ until you find cell $j$: $hid(j) \notin cache$;
 9:             $r \leftarrow hid(j)/8k$;
10:             insert $r$ into $rows[\,]$;
11:         end for
12:         $cells[\,] \leftarrow PIR(rows[\,])$;
13:         $cache \leftarrow cells[\,]$;
14:     end if
15:
16: end procedure
```

Fig. 2.3 illustrates an example of the prediction algorithm for $u = 2$ and $u = 4$. The dots in these figures represent the user's trajectory (starting from the upper left corner), and the shaded boxes represent the rows retrieved from the WSDB. When $u = 2$ (Fig. 2.3a), the first PIR query is constructed by extending the predicted line, until it encounters the cell marked with the hollow square. As a result, the first PIR query retrieves rows 31 and 30. When the user enters row 28, a new query is issued for rows 28 and 4. This process repeats and the user issues a total of four PIR queries, represented by the white dots in Fig. 2.3a.

On the other hand, when $u = 4$ (Fig. 2.3b) the client is able to prefetch more results from the predicted trajectory, thus resulting in just two PIR queries for the entire trajectory. The first query retrieves rows 31, 30, 28, and 4, while the second one retrieves rows 5, 6, 8, and 9. Note that, reducing the number of PIR queries is very important, as they incur a high computational cost at the WSDB. Regarding the communication cost in our example, the 2 sub-segment case requires a total of $40 \cdot \log m$ bits, while the 4 sub-segment case requires $36 \cdot \log m$ bits. In other words, for approximately the same communication cost, we were able to reduce the computational cost at the WSDB by 50% (4 vs. 2 PIR queries).



(a)

(b)

Figure 2.3: Trajectory prediction example. (a) Using 2 sub-segments. (b) Using 4 sub-segments.

2.4.5 A Priori Trajectory Knowledge

Our last method deals with mobile users that have full a priori knowledge of their trajectories. This is not an unrealistic assumption, since that feature is common in GPS navigation systems. In this scenario, users are allowed to choose the trajectory starting and ending points, and then control the route connecting the two end points. Knowing the exact trajectory enables us to simulate the route on the underlying grid, and identify the cells that intersect with that route. Algorithm 4 depicts that simulation. It simply initializes an empty hash table HT, and inserts therein the row numbers of all cells that intersect trajectory T.

Algorithm 4 A Priori Trajectory Simulation

```
procedure A-PRIORI-TRAJECTORY-SIMULATION($T, k$)

    $HT \leftarrow \emptyset$;

    for each cell $i$ intersecting trajectory $T$ do
        $r \leftarrow hid(i)/8k$;
        insert $r$ into $HT$;
    end for

end procedure
```

Once the algorithm computes the final hash table, the client has two options regarding query processing. The first one is to issue $|HT|/u$ PIR queries to the WSDB and retrieve all the necessary results beforehand. The second option is to issue the queries “on-demand.” That is, when the client moves into a cell without any channel availability information, he retrieves the row of that cell as well as $(u-1)$ other rows from the hash table (it could be the ones that are spatially close to the query point).

2.5 Experimental Evaluation

In this section, we evaluate experimentally the performance of our proposed methods. Section 2.5.1 describes the setup of our experiments, and Section 2.5.2 provides the detailed results.

2.5.1 Experimental Setup

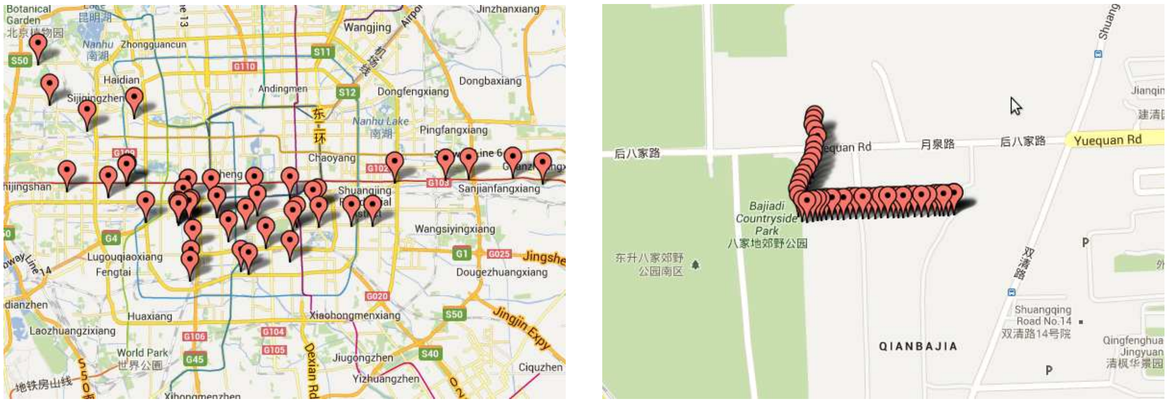
We developed our experiments in Java SDK, running on Ubuntu 14.04.1 LTS. For the experimental tests, we utilized two real life datasets, namely Microsoft's GeoLife GPS Trajectories[1] and Microsoft's T-Drive GPS Dataset[2]. Both are excellent datasets, containing real life trajectories from users traveling around Beijing, China.

The GeoLife GPS trajectory dataset [97] was collected as part of the Microsoft Research Asia GeoLife project. The GeoLife GPS dataset monitors 182 users for a period of over five years (from Apr. 2007 to Aug. 2012). A GPS trajectory from this dataset is represented by a sequence of time-stamped points, each containing information regarding the user's latitude, longitude, and altitude. The dataset contains 17,621 trajectories, with a total distance of 1,292,951 kilometers, and a total duration of 50,176 hours. These trajectories were recorded by different GPS loggers and GPS-enabled phones, and have a variety of sampling rates. 91.5 percent of the trajectories are logged in a dense representation, e.g., every 1–5 seconds or every 5–10 meters per point. A sample trajectory from the GeoLife GPS dataset is depicted in Fig. 2.4b.

The T-Drive dataset [93] contains GPS trajectories from 10,357 taxis, during the period of Feb. 2 to Feb. 8, 2008. The total number of points in the dataset is about 15 million,

[1] `http://research.microsoft.com/en-us/projects/GeoLife/`

[2] `http://research.microsoft.com/en-us/projects/tdrive/`



(a) (b)

Figure 2.4: (b) Sample dense data points from the GeoLife trajectories (a) Sample sparse data points from the T-Drive trajectories

and the total distance from all trajectories reaches up to 9 million kilometers. The average sampling interval is about 177 seconds, with a distance of about 623 meters. A sample trajectory from the T-Drive dataset is shown in Fig. 2.4a.

In our experiments, we set a bounding box of 409.6km × 409.6km (thus setting $n = 4096$) around Beijing's coordinates, which are 39.9139°N, 116.3917°E. The bounding box's coordinates are set as $minlat = 37.7$, $maxlat = 41.5$, $minlong = 114.1$, and $maxlong = 118.9$. From all the available trajectories, we compiled a list of the longest ones that are completely contained within the bounding box. In particular, we selected 9727 trajectories from the T-Drive dataset, and 2188 trajectories from the GeoLife dataset.

As performance metric, we measure the average *cumulative* query response time from all PIR queries that are issued to the WSDB throughout the duration of a mobile user's trajectory. This cost includes (i) the query generation time at the client, (ii) the processing time at the server, (iii) the network transfer time, and (iv) the result extraction time at the client. To provide realistic results, we implemented the underlying PIR protocols ([30] and [86]) using the GMP[3] multiple precision arithmetic library. Table 2.3 shows the detailed costs. The client-side computations are performed on an iPhone 5 device running iOS 7.1,

[3] http://gmplib.org

while the server-side computations are performed on a 3.5 GHz Intel Core i7 processor.

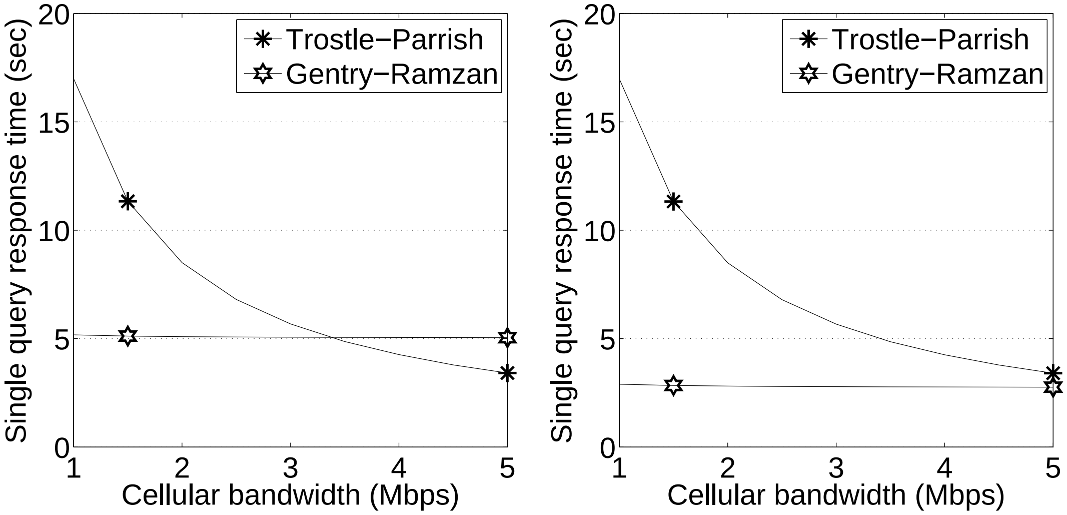
Table 2.3: Cost of PIR operations

Cost	**GR [30]**	**TP [86]**
Query generation (client)	450ms	Pre-processing
Server processing (64 CPUs)	4560ms	18ms
Server processing (128 CPUs)	2280ms	9ms
Result extraction (client)	125ms	0.5ms
Communication cost	20800 bytes	2121728 bytes

For Trostle and Parrish's scheme we set the modulus size equal to 2048 bits, as described in [27]. Note that, the query generation cost at the client can be avoided, since it involves the computation of n random values that are independent of the queried row. As a result, these values can be pre-computed offline. For Gentry and Ramzan's protocol we set the modulus size m equal to 1280 bits, in order to satisfy the security requirement of the protocol.

Also, the query generation algorithm for Gentry and Ramzan depends on the queried row(s) and should be computed online. The values shown in the table above correspond to the single row retrieval method, where $k = 128$ and $u = 1$. While part of the query generation algorithm is precomputed offline (prime q of the RSA modulus), prime p depends on the queried row(s) and should be computed online.

Fig. 2.5 shows the query response time for the two PIR protocols (based on Table 2.3) as a function of the cellular bandwidth available at the mobile client. Clearly, the cost of Trostle and Parrish's scheme is dominated by the network transfer time, since each PIR query necessitates the exchange of over 2 MB of data. On the other hand, Gentry and Ramzan's protocol is practically independent of the available bandwidth, and its cost is determined solely by the computing power at the WSDB (64 vs. 128 CPUs). Nevertheless, as we mentioned earlier, the primary deployment targets for database-driven DSA are areas with scarce cellular bandwidth, making Gentry and Ramzan's protocol a better choice as the underlying PIR mechanism.



(a) (b)

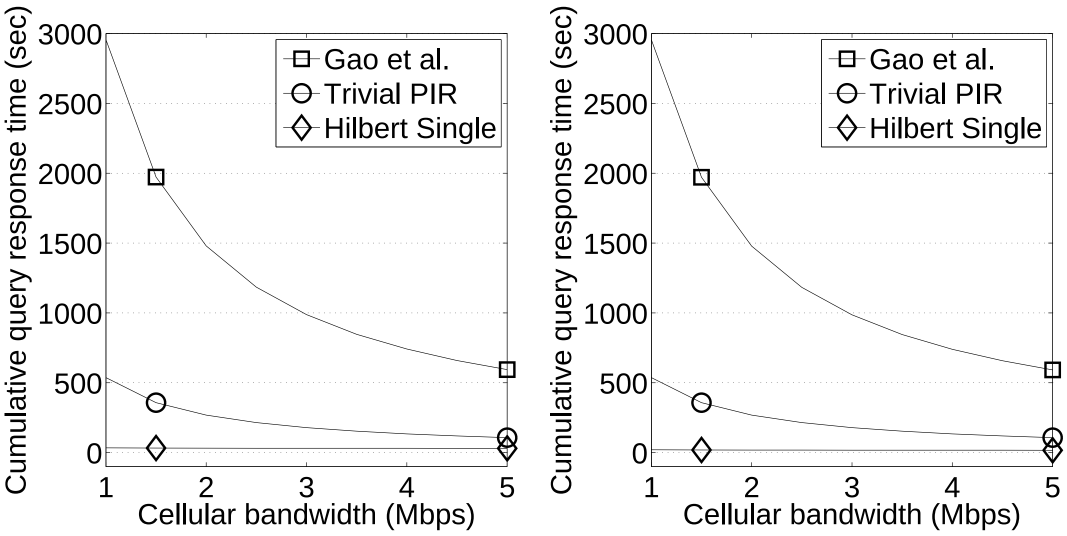
Figure 2.5: Response time for a single PIR query. (a) 64 CPUs (b) 128 CPUs

Note that, another option for achieving location privacy is through the *trivial* PIR case, i.e., by downloading the entire spectrum WSDB with one query. However, this is only viable when the database size is small or when there is ample bandwidth to do so. In our experiments, the database size is over 67 MB, which takes around 536 seconds to download at 1 Mbps, and 108 seconds at 5 Mbps.

2.5.2 Experimental Results

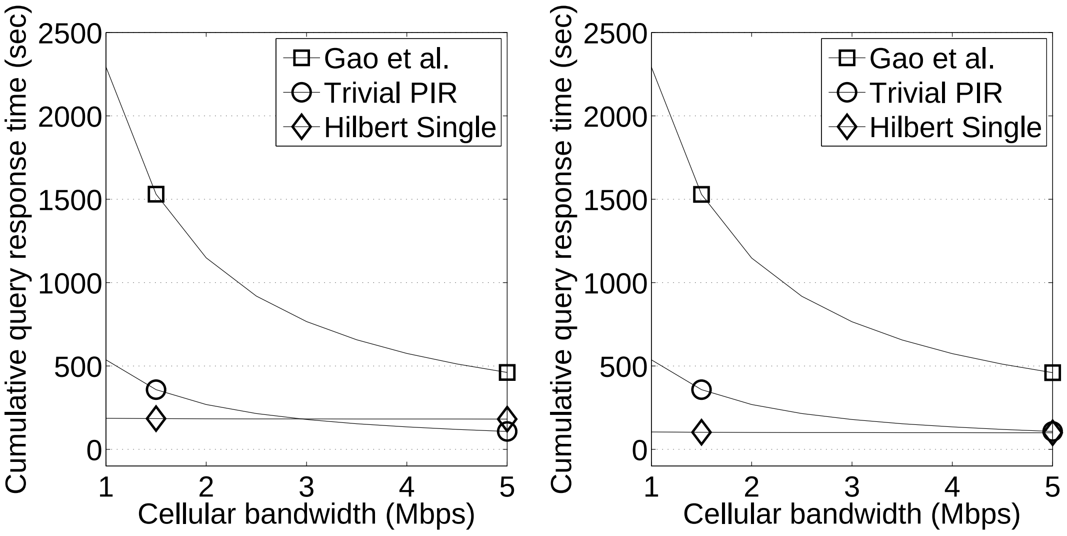
In the first experiment we investigate the performance of the single row retrieval method ($k = 128$, $u = 1$), as explained in Section 2.4.2. Figs.2.6 and 2.7 depict the average cumulative query response time as a function of the available bandwidth.

The curve labeled “Gao et al.” corresponds to an improved version of the original protocol [27] that incorporates Trostle and Parrish's unmodified scheme, which retrieves one row with a single query. (Note that, the two variants have practically identical performance in terms of both computation and communication cost.) Clearly, Gao et al. is not designed for moving clients. In Fig. 2.6 it averages 174 PIR queries (per trajectory) for the GeoLife dataset, and



(a) (b)

Figure 2.6: Response time for the single row retrieval method (GeoLife). (a) 64 CPUs (b) 128 CPUs



(a) (b)

Figure 2.7: Response time for the single row retrieval method (T-Drive). (a) 64 CPUs (T-Drive) (b) 128 CPUs (T-Drive)

135 queries for the T-Drive dataset as shown in Fig. 2.7. As a result, the overall cost of Gao et al. exceeds the cost of the trivial PIR case by a wide margin and we will, thus, omit it from further comparisons in our experiments.

As shown in Figs. 2.6 and 2.7, our single row retrieval method outperforms the trivial PIR case for the GeoLife dataset, and is marginally worse for the T-Drive dataset (for 64 CPUs) when there is adequate download bandwidth. The difference in performance across the two datasets is explained by the structure of the underlying trajectories. Recall that the data points in the T-Drive dataset are recorded at sparse distances (average 623m). The sparseness of the data points mimics well the requirements of a "paging" application, where ubiquitous connectivity is not a requirement. In this scenario, prefetching results from the surrounding area is not always beneficial, since the user may issue the next query from an entirely different area. On the other hand, the data points in the GeoLife dataset are very dense so, with a high probability, several consecutive queries may be issued within a small area.

In the remainder of this section, we investigate the performance of our multi-record retrieval protocols for the case of $k = 128$ and $u = 4$. We begin by evaluating the surrounding area method, which was explained in Section 2.4.3 (we set $R = 50$ rings as the explored area). Figs. 2.8 and 2.9 illustrate the cumulative query response time as a function of the available bandwidth for the two datasets.

It is evident that our method outperforms considerably the trivial PIR case in almost all settings. Compared to the single row retrieval method (which is also included in the figure for clarity), the surrounding area approach decreases the overall query cost by 57%, on average, for the GeoLife dataset as shown in Fig. 2.8. Fig. 2.9 shows that the surrounding area approach decreases the overall query cost by 47%, on average, for the T-Drive dataset.

Next, we evaluate the performance of our trajectory prediction method, as described in Section 2.4.4. Figs. 2.10 and 2.11 show the cumulative query response time for the three

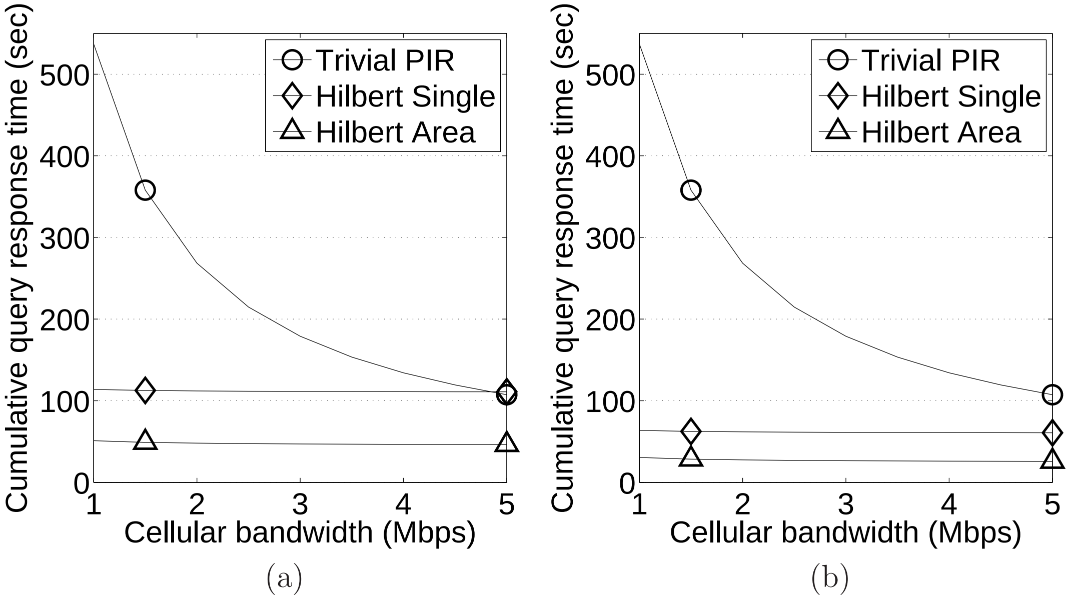


Figure 2.8: Response time for the surrounding area method (GeoLife). (a) 64 CPUs (b) 128 CPUs

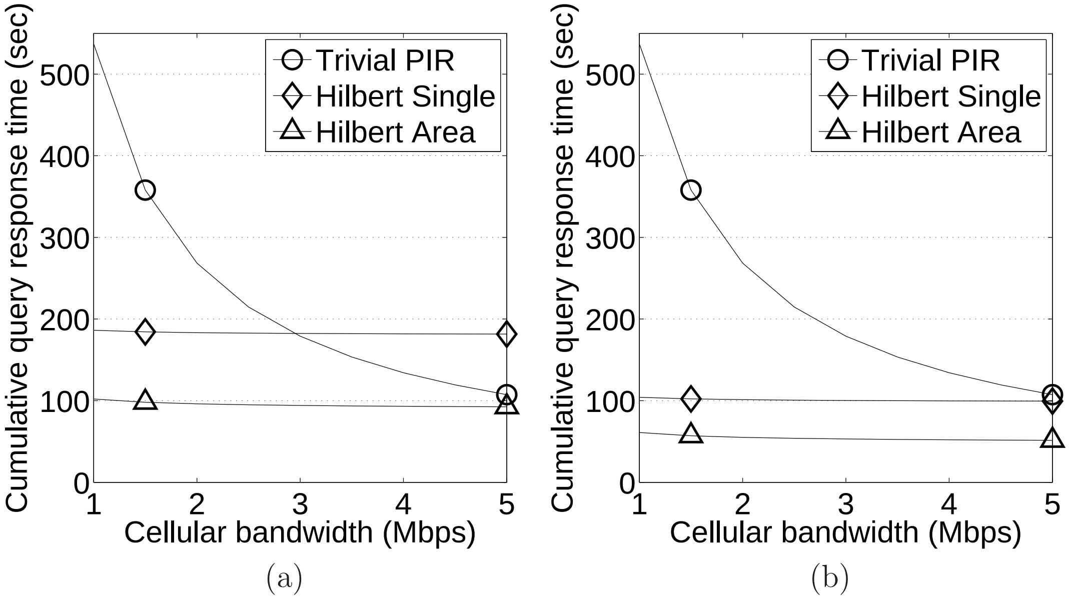
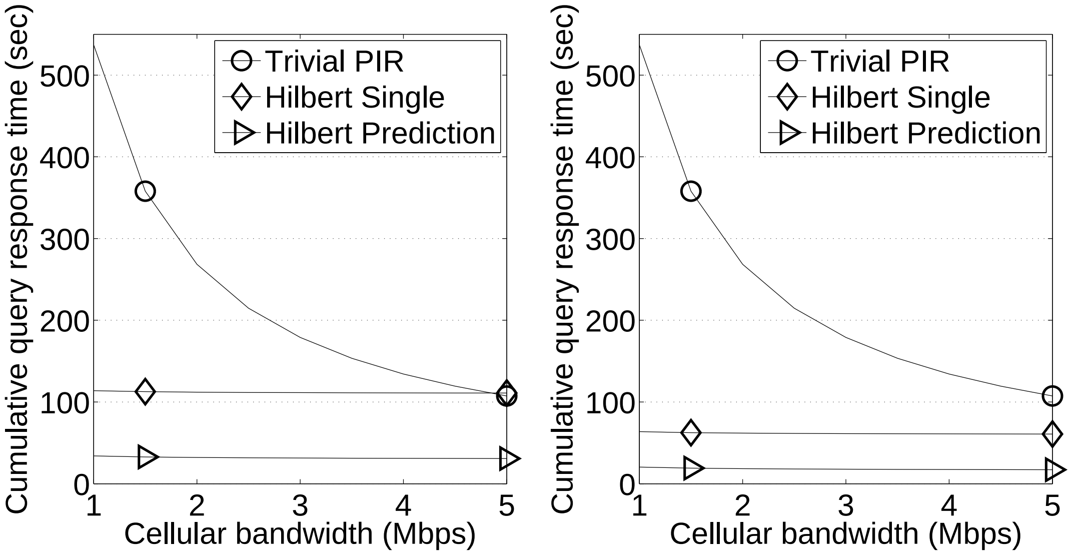


Figure 2.9: Response time for the surrounding area method (T-Drive). (a) 64 CPUs (b) 128 CPUs

methods under various settings.

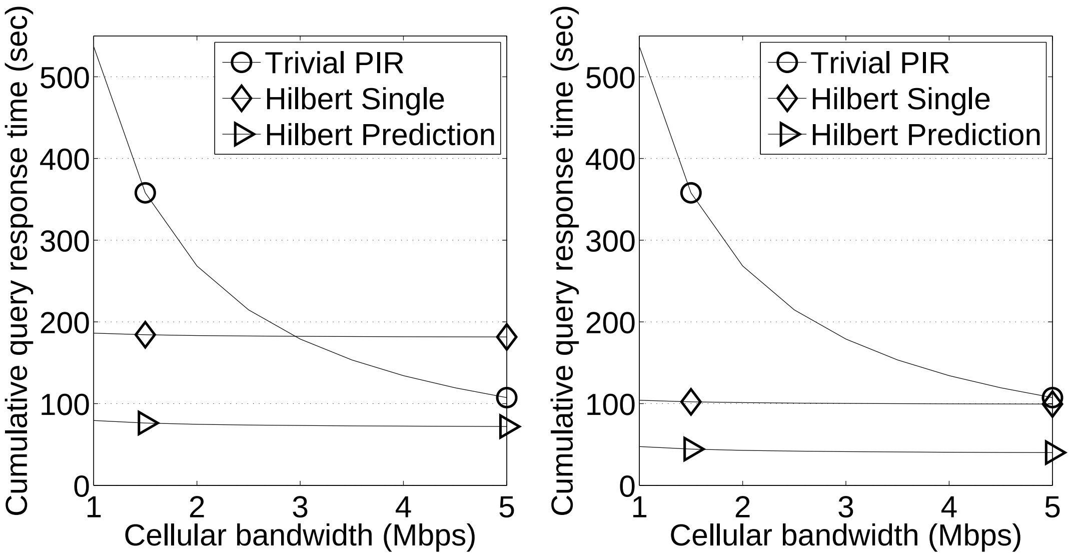


(a) (b)

Figure 2.10: Response time for the trajectory prediction method (GeoLife). (a) 64 CPUs (b) 128 CPUs

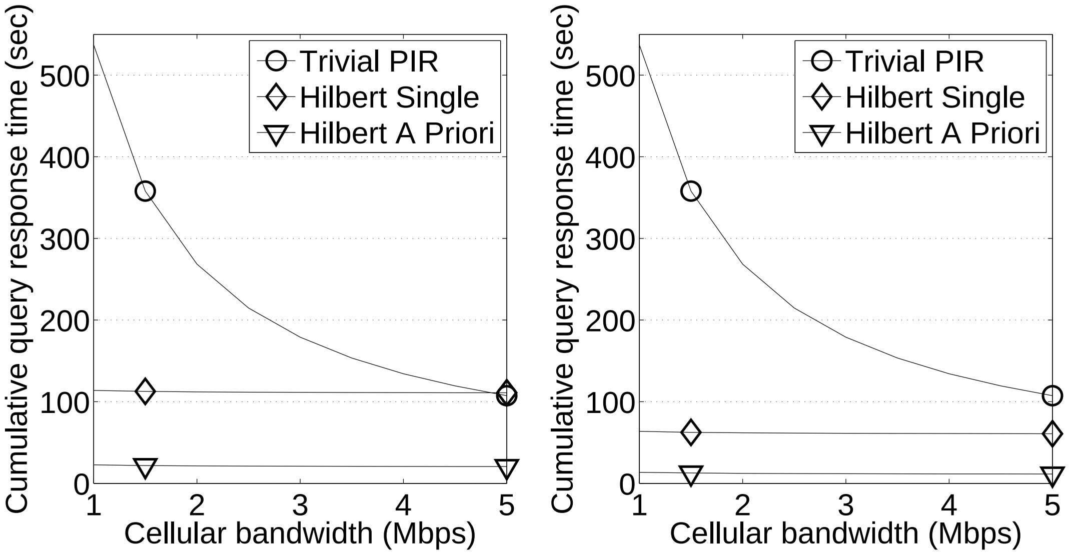
The trajectory prediction approach is clearly superior to the trivial PIR case under all settings. Utilizing 128 compute units at the server results in a response time of just 18 sec (for the whole trajectory) in the GeoLife dataset, as shown in Fig 2.10 and 42 sec in the T-Drive dataset, as shown in Fig 2.11. In addition, the trajectory prediction algorithm decreases the query processing cost even further compared to the surrounding area method. Specifically, it reduces the cost by an additional 34% in the GeoLife dataset, shown in Fig 2.10, and 28% in the T-Drive dataset, as shown in Fig 2.11. This is due to the fact that, with trajectory prediction, we prefetch results according to a specific direction of movement instead of a generic rectangular area.

In our last experiment, we investigate the performance of the a priori trajectory knowledge approach, which was explained in Section 2.4.5. This method yields the lowest PIR requests, since the client avoids the retrieval of any unnecessary rows from the WSDB. Figs. 2.12 and 2.13 illustrates the corresponding cumulative query response times.



(a) (b)

Figure 2.11: Response time for the trajectory prediction method (T-Drive). (a) 64 CPUs (b) 128 CPUs



(a) (b)

Figure 2.12: Response time for the a priori trajectory knowledge method (GeoLife). (a) 64 CPUs (b) 128 CPUs

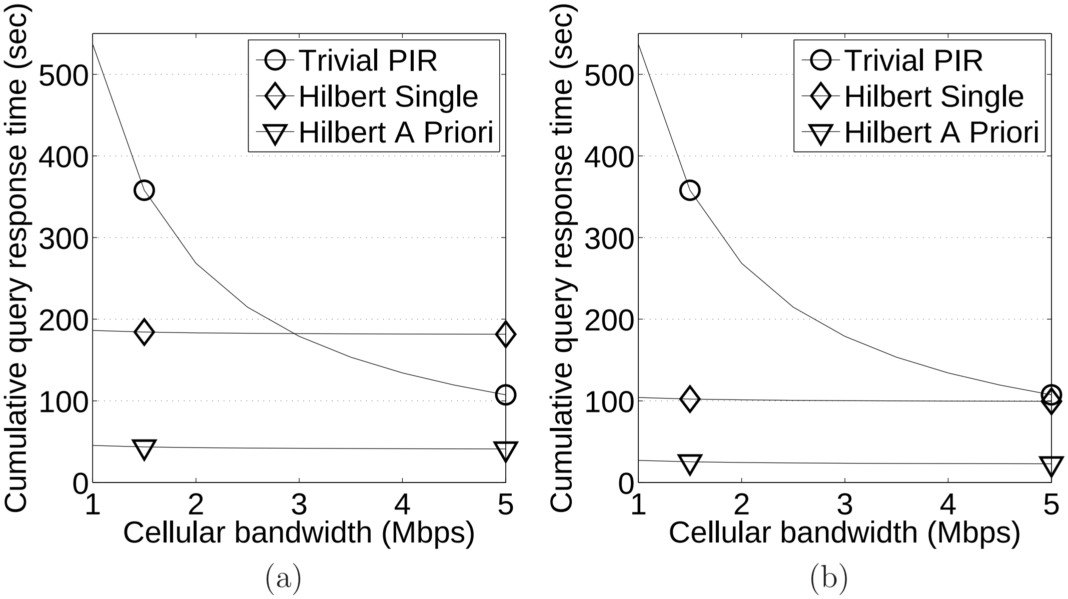


Figure 2.13: Response time for the a priori trajectory knowledge method (T-Drive). (a) 64 CPUs (b) 128 CPUs

Our method outperforms the trivial PIR approach by a large factor, and entails a cost of just 12 sec in the GeoLife dataset, as shown in Fig. 2.12, and 23 sec in the T-Drive dataset (for 128 compute units), as shown in Fig. 2.13. Compared to the trajectory prediction method, the a priori trajectory knowledge enables us to reduce the query processing cost by an additional 33% in the GeoLife dataset, as shown in Fig. 2.12, and 43% in the T-Drive dataset, as shown in Fig. 2.13.

2.6 Summary

Existing methods for location privacy in the database-driven DSA model are very inefficient, because they are not optimized for mobile clients. To this end, the work presented in this chapter introduces an efficient solution, based on a Hilbert space filling curve indexing of the white-space database. Our methods leverage a communication-efficient PIR protocol, and employ trajectory prediction algorithms to minimize the number of PIR queries. Through extensive experimentation with real life datasets, we show that, compared to the current state-of-the-art protocol, our methods reduce the query response time at the mobile clients by a large factor.

CHAPTER 3

P2P INTERACTIONS FOR EFFICIENT LOCATION PRIVACY IN DSA

Our work on P2P interactions towards efficient location privacy in DSA is organized as follows. Section 3.1 provides the necessary background on the cryptographic primitives utilized in our methods. Section 3.3 describes the details of our P2P protocol and Section 3.4 presents the results of the experimental evaluation.

3.1 Preliminaries

Here, we give a brief description of the cryptographic primitives incorporated in our methods. Section 3.1.1 provides some background on anonymous communication and Section 3.1.2 introduces the 2-round anonymous veto network (AV-net) of Hao and Zieliński [39].

3.1.1 Anonymous Communication

Research on anonymous communication has evolved due to the *dining cryptographers* problem, introduced by Chaum in 1988 [12]. Essentially, a dining cryptographers network (DC-net) allows groups of $n > 2$ participating users to contribute their boolean bits towards a boolean-OR calculation of some statement, while preserving the privacy of the individual inputs. DC-nets have many weaknesses and are considered impractical due to complex key setup, message collisions, and vulnerability to disruptions. Alternatively, circuit evaluation techniques, such as the ones proposed in [32, 92], can also be used towards the secure computation of a boolean-OR function. However, as pointed out by Brandt [6], the circuit evaluation technique is expensive and impractical.

A similar problem is the anonymous veto network (AV-net), which allows groups of $n > 2$ participating users to vote against a given statement. In the setting of a white-space TV bands database, where channel availability can be represented via a boolean bit, a sample statement might be: “none of the group members knows that the channel is free.” If any of the users in the group anonymously vetoes the statement, it means that “at least one of the users in the group knows that the channel is free.”

Unlike DC-nets, AV-net protocols do not require secret channels in order to exchange messages. Furthermore, they have no message collisions and are very resistant to disruptions. Nevertheless, all existing AV-net protocols assume the existence of an *authenticated broadcast channel*, which is easily implemented using digital signatures [12]. Several such anonymous veto protocol designs exist in the literature [6, 34, 39, 51]. In our work, we leverage the 2-round AV-net protocol of Hao and Zielinśki [39], because it is more efficient in terms of number of rounds, computation, and communication cost.

3.1.2 2-Round AV-net Protocol

Setup. All users participating in the protocol agree on two public parameters, namely G and g. G is a finite cyclic group of prime order q in which the Decision Diffie-Hellman (DDH) problem is hard [4], and g is a generator of G. These values are fixed and used in all protocol invocations. Subsequently, each participant P_i, $i \in \{1, 2, \dots, k\}$, selects a random secret value $x_i \in_R \mathbb{Z}_q$.

Round 1. In the first round, every participant P_i broadcasts g^{x_i}. When the first round completes, each participant P_i computes

$$g^{y_i} = \prod_{j=1}^{i-1} g^{x_j} \Big/ \prod_{j=i+1}^{k} g^{x_j}$$

Round 2. In round 2, every participant P_i broadcasts a value $g^{c_i y_i}$ where c_i is either x_i or a random value $r_i \in_R \mathbb{Z}_q$, depending on whether participant P_i vetoes the statement or not.

$$g^{c_i y_i} = \begin{cases} g^{r_i y_i} & \text{if } P_i \text{ sends `1' (veto),} \\ g^{x_i y_i} & \text{if } P_i \text{ sends `0' (no veto).} \end{cases}$$

In order to test the final result, all participants compute $\prod_i g^{c_i y_i}$. If nobody vetoed the statement, then $\prod_i g^{c_i y_i} = \prod_i g^{x_i y_i} = 1$, since $\sum_i x_i y_i = 0$. If, however, one or more participants vetoes the statement by sending a '1', we have $\prod_i g^{c_i y_i} \neq 1$.

3.2 Threat Model and Security

In this work, we are not concerned with privacy against the WSDB operator. We assume that mobile users, when needed, query the WSDB through a standard PIR protocol. Instead, in our methods, the adversary is one or more users in the group that executes the AV-net protocol, or any eavesdropper that monitors the exchange of messages over the wireless channel. The adversary runs in polynomial time, and its goal is to identify a user that vetoes a certain statement.

Note that, in the case of malicious adversaries, the protocol described above necessitates zero-knowledge proof (ZKP) schemes, such as Schnorr's signature [74]. In particular, during each round, users must demonstrate knowledge of their own secret values, such as x_i and c_i. Nevertheless, in our work, we assume the *honest-but-curious* adversarial model, i.e., all users follow the protocol correctly but try to gain an advantage by examining the communication transcript. As a result, we do not implement zero-knowledge proofs.

Our methods inherit the security of the underlying AV-net protocol [39]. As such, they are *semantically secure* [33], i.e., it is infeasible to derive any information about a mobile client's input, given its published values and the public parameters. The security is based on the DDH assumption. Therefore, an eavesdropper is unable to determine whether a user has vetoed a statement. Our methods are also secure against *partial* collusions, i.e., when some participants collude (by revealing their secret values) to determine the input of a certain user. As explained in [39], only a *full* collusion against a single user can compromise security, i.e., when $k-1$ users reveal their values to identify whether the k-th user vetoed a statement.

3.3 P2P Protocol

In this section, we present the P2P protocol that allows a group of users to share anonymously their cached spectrum information. Section 3.3.1 describes the system architecture, and Section 3.3.2 explains the protocol initiation process. Section 3.3.3 presents the criteria for mobile nodes to participate in this protocol, and Section 3.3.4 describes the group formation mechanism. Finally, Section 3.3.5 introduces the details of the AV-net protocol invocation.

3.3.1 System Architechture

Similar to previous work [27], we assume a fixed grid of $n \times n$ cells, where mobile users can communicate through white-space TV bands, while maintaining their location privacy. According to the FCC specifications [22], each cell is 100m×100m in size, and users may need to query the WSDB whenever they move into a cell with no prior spectrum availability knowledge. The dimensions of the grid (i.e., n) can be made arbitrarily large, which has a direct effect on the database size.

Note that, mobile TVBDs are allowed to communicate only in the frequency ranges 512-608 MHz (TV channels 21-36) and 614-698 MHz (TV channels 38-51), i.e., there are a total of 31 possible white-space TV band channels that can be accessed in a DSA manner. Therefore, we represent the daily channel availability as 32 bits (per cell), where bit 0 represents a busy channel and bit 1 represents an idle channel. As an example, when $n = 5000$, the WSDB is 100 MB in size. The trivial PIR case is impractical in this setting, since it involves downloading the entire WSDB. This would take approximately 35 mins on 3G networks[1] [78].

In our model, we assume an out-of-band common control channel (CCC) through a dedicated transceiver. This enables mobile users to exchange concurrently both control and data messages. Out-of-band CCC coordination can be realized over the 802.11 protocol in *ad-hoc* mode or through any of the methods proposed in [2, 16]. We emphasize that 802.11 is not a viable protocol for long range communications, hence it is only used to implement the out-of-band CCC for communications within a 100m×100m cell.

The FCC's white-space TV band DSA specifications state that "A mode II personal/-portable device may load channel availability information for multiple locations around, i.e., in the vicinity of, its current location and use that information in its operation." Accordingly, in our methods, we assume a PIR protocol that retrieves channel information for multiple cells with a single query[2]. As a proof of concept, we consider a fixed grouping of the available cells into 4×4 blocks. Therefore, we assume that each PIR query retrieves the 16-cell block that contains the user's current cell.

Fig. 3.1a shows an example of this approach. The black colored cells signify the locations where a new PIR query is issued, due to lack of spectrum availability knowledge. The alternating white and grey colored cells identify the different blocks, with the block *id* shown

[1] Furthermore, communication over a cellular network is a priced resource that should be avoided.

[2] All existing PIR schemes can retrieve multiple records from a database.

in the lower-left corner of the block. Note that, even though we assume a specific method for querying the WSDB, our protocol is *orthogonal* to the underlying PIR query/reply structure. Any WSDB indexing method is a viable candidate for our protocol, but for the PIR reply to be of some utility to the client, the retrieved cells should be spatially close to the user's location.

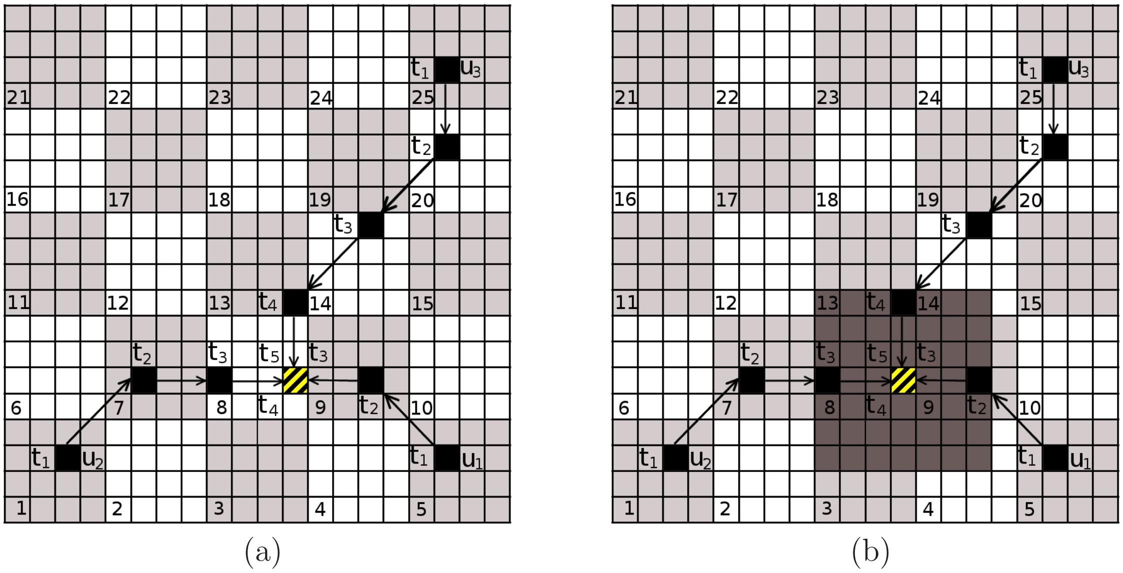


Figure 3.1: (a) Three mobile users querying a WSDB via PIR, and intersecting at the diagonally striped cell (b) Three mobile users invoking the AV-net protocol for the region identified by the darker shaded cells

As illustrated in Fig. 3.1a, each of the three mobile TVBDs gradually builds a spectrum knowledge *cache* containing channel availability information from their respective trajectories. When the users eventually meet at the diagonally striped cell, it may be beneficial to all of them to exchange their cached information. To maximize utility for all participating users, the sharing of spectrum information involves the area surrounding the current location (as users may continue their trajectories towards any direction). In particular, the TVBD nodes agree on the number of surrounding rings (AR) that they wish to explore during the protocol invocation. (Table 3.1 summarizes the symbols used in the remainder of this paper, along with the values tested in the experimental evaluation.) In the example of Fig. 3.1b,

$AR = 3$, and the explored region is shown in a darker shade.

Table 3.1: Summary of Symbols

Symbol	**Description**	**Range**
GS	Group size	3-10
BS	Number of cells in a PIR block	16
AR	Ring(s) explored through AV-net invocations	1-3
AP	AV-net participation probability (fixed)	0-1
PI	AV-net participation probability increment (TCP)	0.05-0.2
K	AV-net initiation threshold	0.2-0.8
AK	Actual knowledge of the AR area	0-1

To illustrate the location privacy leakage from a *plaintext* exchange of spectrum availability information (i.e., without the invocation of the AV-net protocol), consider the example of Fig. 3.1b. We can infer that u_1 arrived at the current cell through block 9, while u_3 visited blocks 13 and 14. On the other hand, u_2's trajectory contains some uncertainty, as u_2 may have arrived at the current cell through blocks 2, 7, or 12. Furthermore, if two users participate in the same group multiple times (at different locations), they can derive more information about each other's movement patterns.

We assume that intersecting users remain within communication range for ample periods of time (e.g., 1-2 minutes). However, they do not need to reside in the same cell continuously. The three conditions that control a successful invocation of our protocol are (i) protocol *initiation*, (ii) protocol *participation*, and (iii) successful *group formation*. Group formation is dependent on at least three users willing to engage in the P2P protocol, such that at least one of the engaging users is an *initiator*. We examine each condition separately in the following sections.

3.3.2 Protocol Initiation

Ideally, a mobile TVBD would like to maintain DSA connectivity throughout its trajectory, without any disruptions. As such, whenever the TVBD moves into a new cell, it measures the ratio of knowledge (AK) in the surrounding area. If that ratio falls under a system-defined threshold K, it initiates the protocol that triggers the group formation algorithm (described later). Algorithm 5 shows the detailed protocol initiation procedure. If there is no channel availability information for the current cell, the user always initiates the protocol (lines 6–8), because it needs to identify free channels. On the other hand, if the current cell does exist in its cache, it computes the ratio AK for the present position (lines 9–15). Specifically, each surrounding ring is assigned an identical aggregate weight (equal to $1/AR$), which is split equally among the individual cells. As a result, cells in the inner rings carry more weight than those in the outer rings, and lack of knowledge in the inner rings is more likely to initiate the protocol.

Algorithm 5 Protocol initiation Algorithm

```
 1: procedure INITIATE-PROTOCOL
 2:
 3:     bool $initiate$ = false;
 4:     double $AK = 0.0$;
 5:
 6:     if no spectrum information for current cell then
 7:         $initiate$ = true;
 8:     else
 9:         for $i = 1$ to $AR$ do
10:             for all cells $c_j$ in ring $i$ do
11:                 if no spectrum information for $c_j$ then
12:                     $AK = AK + 1.0/(AR \cdot 8 \cdot i)$;
13:                 end if
14:             end for
15:         end for
16:         if $AK \leq K$ then
17:             $initiate$ = true;
18:         end if
19:     end if
20:     return $initiate$;
21: end procedure
```

3.3.3 Protocol Participation

Participation is defined as the selfless event, where one or more users in the group decide to participate in the AV-net protocol for the purpose of disseminating (and also collecting) channel information about the surrounding area. In order to avoid meaningless (due to repetition) AV-net protocol invocations that could lead to battery drainage, we propose the following three probabilistic AV-net participation methods.

Fixed probability. This is the simplest approach where, whenever a protocol is initiated, a nearby TVBD always chooses to participate with probability AP. Larger AP values produce a greedy behavior that is optimal in terms of PIR query savings. On the other hand, this may also lead to numerous AV-net invocations in close (spatial) proximity, which are redundant in terms of gained knowledge.

TCP-like approach. In the second method, we borrow from TCP Reno's congestion control mechanism [46]. In particular, a mobile user starts with a participation probability $AP = 1.0$. At each successful AV-net participation, AP is cut by half. Otherwise, if there is a protocol initiation but the TVBD does not participate, AP is incremented by PI units. This technique is expected to be the most conservative one, due to its aggressive back-off behavior.

Weighted sliding window. The final method is based on the weighted sliding window (SW) projection. We experimented with different window sizes, and decided to utilize a model with five entries, such that $W_1 = 0.5$, $W_2 = 0.25$, $W_3 = 0.15$, $W_4 = 0.07$, $W_5 = 0.03$, and $\sum_i W_i = 1$. (W_1 corresponds to the most recent entry.) The current window snapshot is stored as a 5-bit array, where '0' represents participation and '1' represents non-participation. In order to determine the probability of participation, a mobile user first checks its window and sums up past events for which it did not participate. For example, if the current SW snapshot is $(0, 1, 0, 1, 0)$, the user will participate with probability $0.25 + 0.07 = 0.32$. The weighted SW allows us to weight recent historical data more heavily than older ones, when determining the projected probability. This fits well with the intended participation model, in which more recent participation should lead to lower participation probability in the near future.

3.3.4 Group Formation

When Algorithm 5 (protocol initiation) returns *true*, the underlying TVBD initiates an invocation of the AV-net protocol. This is done by broadcasting its interest in the lowest out-of-band CCC channel. Assuming 801.11 as our out-of-band CCC implementation, any potential initiators will broadcast their unique MAC addresses, their current cell id, their *rendezvous* channel id[3], and an initiation flag over 802.11 channel 1. Users that are already engaged in an AV-net invocation/transmission will not hear such broadcast. We assume standard 802.11 MAC contention mechanisms are in place. We coin as “root” the first mobile user that successfully broadcasts the AV-net initiation control packet, regarding a specific cell id. Any other users (including other potential initiators situated in the same cell) that receive the first successful broadcast from a root node, and whose cell id *matches* the broadcast cell id, will use a simple three-way handshake group formation protocol.

Mobile users that decide to participate (based on the methods described earlier) or had attempted to initiate an AV-net invocation themselves, will first switch to the rendezvous channel. They will announce to the root user, through broadcast communication, their willingness to engage. We coin as “children” any of the users that have successfully rendezvoused in the channel id specified by the root user. The three-way handshake broadcast MAC protocol is summarized in Algorithm 6.

[3] Assuming there is a free channel in the out-of-band CCC range.

Algorithm 6 Group formation Algorithm

```
procedure Three-Way-Handshake

    [all children broadcast] Send Request To Join Group

    while group size $< GS$ do
        [root] randomly pick a child
        [root] send Clear to Join Group
        [root] increment group size counter
    end while

    for all  other children who sent a Request to Join  do
        [root] send Reject to Join Group
    end for
    [all who received Clear to Join Group] Send Confirm To Join Group

    [root] send ABORT if group size counter $\leq 3$
end procedure
```

3.3.5 2-Round Protocol Invocation

When a group is formed, the nodes therein execute the AV-net protocol (as described in Section 3.1.2) for each bit of information that they want to share. However, to avoid excessive network delays due to the 2-round nature of the AV-net protocol, we group all individual invocations into two aggregate rounds, as shown in Algorithm 7. Specifically, the users first agree on the the specific order in which the cell information is transmitted, and then each user broadcasts its aggregate data to the rest of the group. The broadcast order can be arranged based on the unique MAC addresses of the TVBDs.

Lines 3–8 (Algorithm 7) correspond the the first round of the AV-net protocol, i.e., each node broadcasts a unique key for every bit of information in the surrounding AR rings. In the example of Fig. 3.1b, where $AR = 3$, each node computes and broadcasts $32 \cdot (1 + \sum_{i=1}^{3} 8 \cdot i) = 1568$ modular exponentiation results. In Round 2 of the protocol (lines 9–23), users publish their spectrum knowledge by choosing the appropriate values for c_{i_b} (as

Algorithm 7 AV-net Protocol

```
 1: procedure AV-NET($G, g$)
 2:
 3:     for all users $i$ in the group do
 4:         for all bits $b$ in the explored area do
 5:             compute $g^{y_{i_b}}$;
 6:         end for
 7:     end for
 8:
 9:     for all users $i$ in the group do
10:         for all bits $b$ in the explored area do
11:             compute $g^{c_{i_b} y_{i_b}}$;
12:         end for
13:         broadcast all exponentiations for user $i$;
14:     end for
15:
16:     for all users $i$ in the group do
17:         for all bits $b$ in the explored area do
18:             compute $r_b = \Pi_i g^{c_{i_b} y_{i_b}}$;
19:             if $r_b \neq 1$ then
20:                 mark the corresponding channel as free;
21:             end if
22:         end for
23:     end for
24: end procedure
```

explained in Section 3.1.2). Specifically, if the underlying channel if free, the user vetoes that particular statement. Note that, in our running example, this step also involves the computation and broadcast of 1568 modular exponentiations. The result extraction phase of the algorithm (lines 16–23) necessitates only GS modular multiplications per bit, and it is optional, i.e., it can be computed only when the user moves into the corresponding cell.

3.4 Experimental Evaluation

In this section we evaluate experimentally the performance of our methods. Section 3.4.1 describes the experimental setup and Section 3.4.2 presents our results.

3.4.1 Experimental Setup

We developed our experiments in Java SDK, running on a Ubuntu 10.4 LTS machine. To simulate the mobile TVBD users, we utilized Microsoft's GeoLife GPS Trajectories[4], which is an excellent dataset containing real-life trajectories from users traveling around Beijing, China. The GeoLife dataset [97] was collected as part of the Microsoft Research Asia GeoLife project, by monitoring numerous users for a period of over five years (from Apr. 2007 to Aug. 2012). A GPS trajectory from this dataset is represented as a sequence of time-stamped points, each containing information regarding the user's latitude, longitude, and altitude.

The dataset includes 17,621 trajectories, with a total distance of 1,292,951 kilometers, and a total duration of 50,176 hours. These trajectories were recorded by different GPS loggers and GPS-enabled phones, and have a variety of sampling rates. More specifically, 91.5 percent of the trajectories are logged in a dense representation, e.g., every 1–5 seconds or every 5–10 meters per point. We randomly selected 2774 intersecting trajectories, each simulating a unique user. For each trajectory, we measure (i) the average number of PIR queries issued by the user, and (ii) the average number of AV-net invocations that the user participates in.

In addition to the simulation results, we also implemented the basic cryptographic operations of the AV-net protocol on an iPhone 5, running iOS 7.1. Specifically, we cross compiled the GMP[5] multiple precision arithmetic library for the ARM architecture, and built a benchmark app to measure the cost of these operations on a handheld device. We generated a cyclic group G of prime order q, where q is a 160-bit number. The group modulus was chosen as a 64-byte prime. Table 3.2 shows the cost of these operations.

[4] `http://research.microsoft.com/en-us/projects/GeoLife/`

[5] `http://gmplib.org`

Table 3.2: Cost of cryptographic primitives

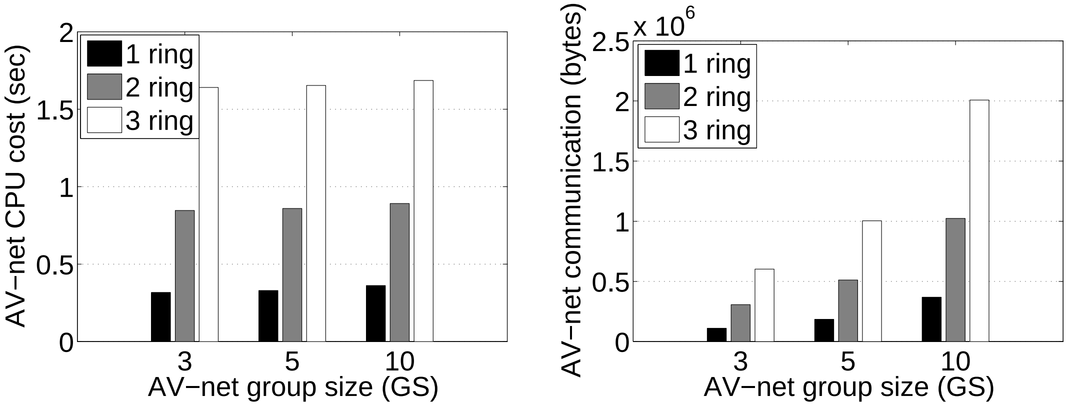
Operation	Cost
Modular multiplication	0.004 ms
Modular exponentiation	0.518 ms

3.4.2 Experimental Results

Fig. 3.2a illustrates the projected CPU time needed to run the AV-net protocol (Algorithm 7) on a handheld device. This cost is dominated by the expensive modular exponentiation operations and is, thus, unaffected by the group size GS. The major factor that determines this cost is the number of surrounding rings (AR) that are explored during a protocol invocation, since each cell contributes 32 modular exponentiations. Nevertheless, even for a value of $AR = 3$, the total CPU time is around 1.65 sec, which is an acceptable cost.

Furthermore, this cost can easily be reduced by 50%, using offline computations. Observe that, during the first round of the AV-net protocol, each node computes and publishes a large number of modular exponentiations. These values do not require any input from the other participating nodes and may, thus, be pre-computed offline. Specifically, a large pool of values (e.g., several hundred thousands) can be computed either at the mobile device during night time (when charging), or at a desktop machine for faster computations. The storage space required to maintain these values is insignificant compared to the storage capabilities of modern handheld devices.

Fig. 3.2b shows the total number of bytes that are broadcast during an AV-net protocol invocation. Clearly, the communication cost is linear in GS, as each group member needs to broadcast its own input to the protocol. We believe that $GS = 5$ is a very reasonable value for anonymity purposes, in which case the communication cost remains below 1 MB. While this cost might appear significant, we stress that, AV-net broadcasts occur over the 802.11 CCC band and do not involve the cellular network infrastructure.



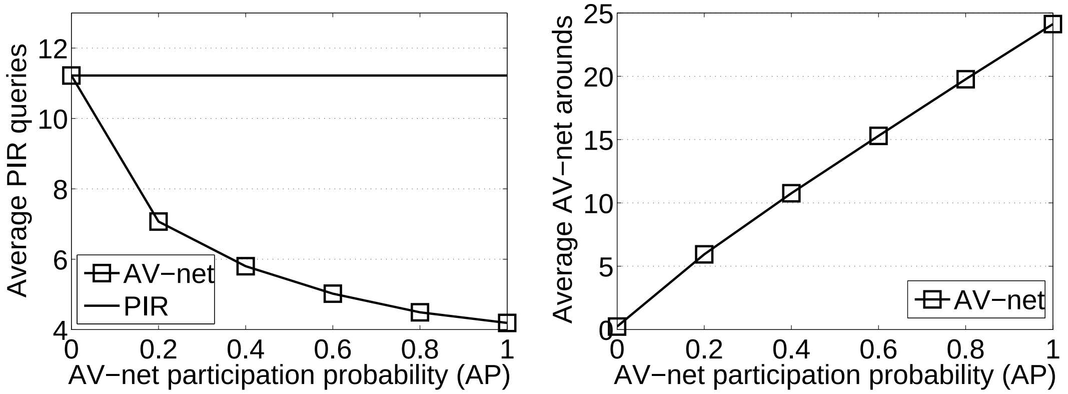
(a) (b)

Figure 3.2: Cost of AV-net protocol on handheld devices (a) CPU cost (b) Communication cost

Fig. 3.3 investigates the effect of the fixed AV-net participation probability (AP) on the performance of our methods. For this experiment, we set $AR = 2$, $GS = 5$, and $K = 0.5$. The curve labeled "PIR" (Fig. 3.3a) corresponds to the PIR-only approach, i.e., when users do not leverage our P2P protocol. When $AP = 0.5$, we observe a 50% reduction in the amount of PIR queries that are sent to the WSDB provider. Larger values naturally lead to better performance (over 60% reduction), but they increase considerably the number of AV-net invocations per user (Fig. 3.3b). Nevertheless, as we have explained previously, PIR queries are much more expensive compared to the AV-net protocol.

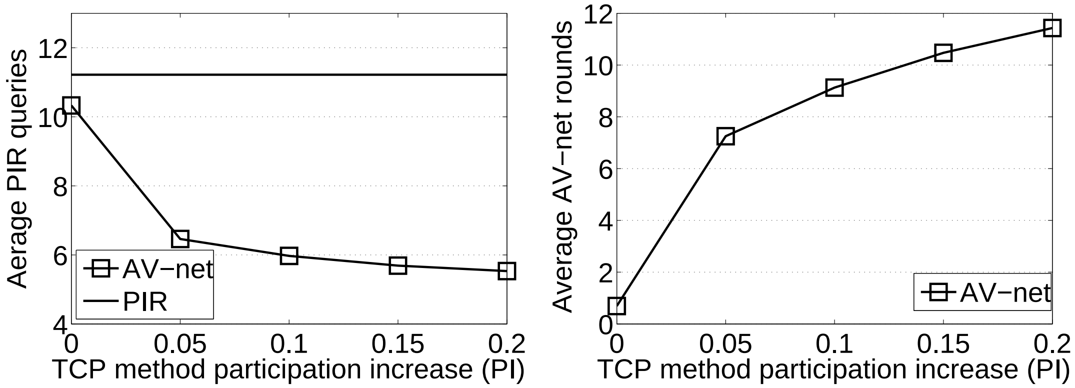
Fig. 3.4 shows the effect of the participation probability increment (PI) for the TCP-like approach ($AR = 2$, $GS = 5$, $K = 0.5$). Lower values of PI discourage users from participating in AV-net protocols and, thus, incur less cost compared to the fixed probability method (Fig. 3.4b). However, as evident in Fig. 3.4a, the TCP-like approach can still reduce the number of PIR queries by up to 50%.

Fig. 3.5 demonstrates the effect of the group size (GS) on the different methods ($AR = 2$, $K = 0.5$, $PI = 0.1$). As Fig. 3.5a implies, larger groups do not contribute more information during the P2P data exchange. Therefore, the average number of PIR queries remains fairly



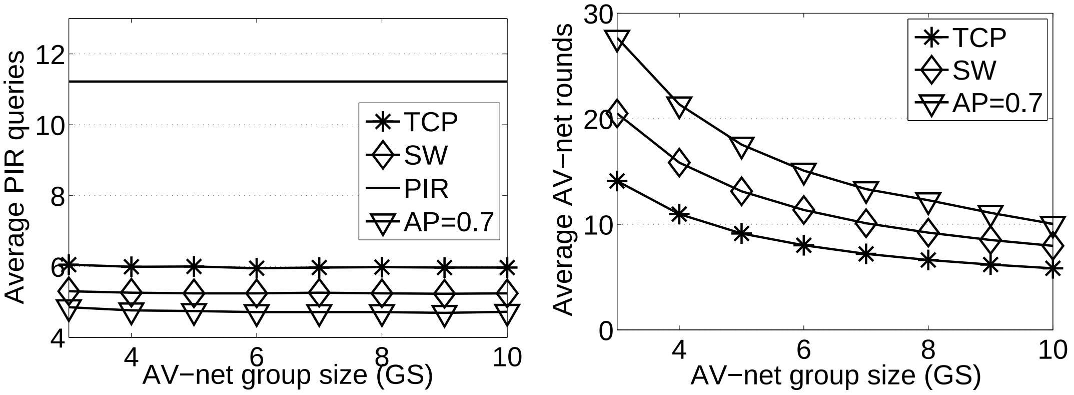
(a) (b)

Figure 3.3: Effect of varying the AV-net participation probability (a) Average number of PIR queries (b) Average number of AV-net invocations



(a) (b)

Figure 3.4: Effect of varying the AV-net participation probability increment (TCP) (a) Average number of PIR queries (b) Average number of AV-net invocations



(a) (b)

Figure 3.5: Effect of varying the AV-net group size (a) Average number of PIR queries (b) Average number of AV-net invocations

constant. Nevertheless, users may still opt for larger groups, in order to gain more privacy. On the other hand, a larger group size reduces the number of AV-net invocations (Fig. 3.5b), because some groups may fail to form due to insufficient number of members. Among the three participation algorithms, the sliding window (SW) approach strikes a good balance between PIR savings (53%) and AV-net overhead (13 rounds, for $GS = 5$).

Fig. 3.6 depicts the effect of the protocol initiation threshold (K) on the different methods ($AR = 2$, $GS = 5$, $PI = 0.1$). Recall that, this threshold represents a lower bound on the amount of spectrum knowledge that a mobile user must possess (regarding the surrounding area), in order to defer an AV-net protocol initiation. As evident in this figure, a knowledge of around 40%-50% is sufficient in terms of overall performance. Larger values to not offer much in terms of PIR reduction, but instead lead to unnecessary AV-net rounds. Similar to Fig. 3.5, the SW participation method has the best performance.

Finally, Fig. 3.7 illustrates the effect of the number of surrounding rings (AR) that are explored during an AV-net protocol invocation ($K = 0.5$, $GS = 5$, $PI = 0.1$). The first observation, is that the number of PIR queries remains almost constant (Fig. 3.7a). The reason is that, as shown in Fig. 3.7b, exploring one ring at a time merely results in more

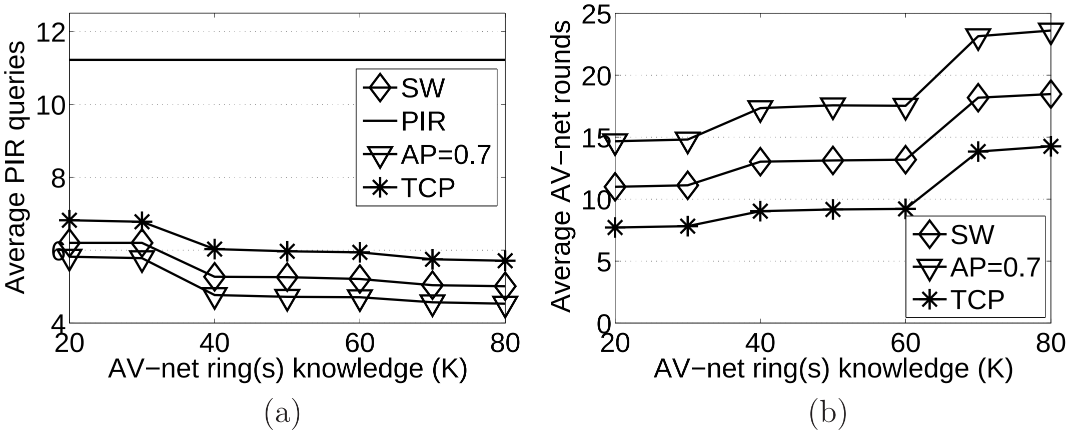


Figure 3.6: Effect of varying the AV-net initiation threshold (a) Average number of PIR queries (b) Average number of AV-net invocations

AV-net rounds, since users invoke a new AV-net protocol once they move further away from their current position. However, the overall PIR reduction is not affected, because users still get most of their spectrum knowledge from the P2P protocol. A value of $AR = 2$ seems like the best choice, given that the number of AV-net rounds does not decrease significantly from 2 to 3 rings.

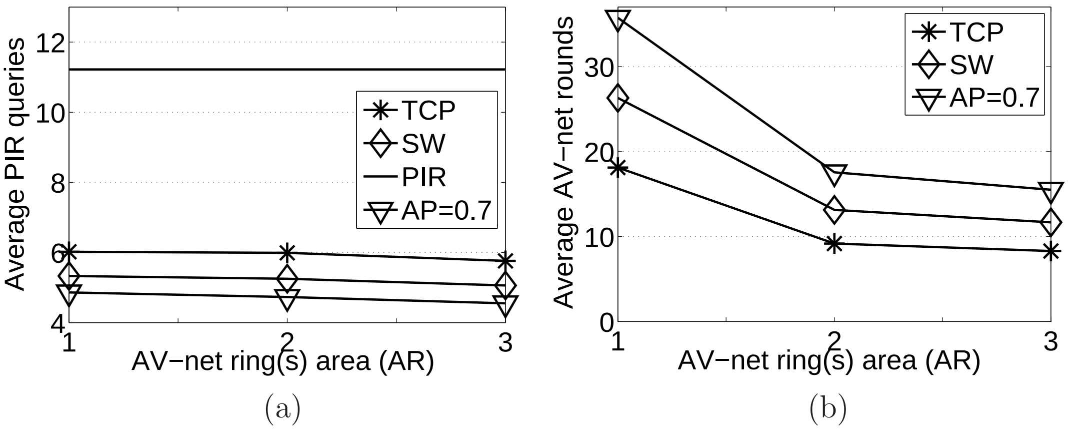


Figure 3.7: Effect of varying the AV-net exploration area (a) Average number of PIR queries (b) Average number of AV-net invocations

3.5 Summary

Database-driven dynamic spectrum access is the standard mode of operation for cognitive radios in the white-space TV bands. This method requires mobile devices to periodically send their location to a centralized white-space database, in order to receive channel availability information in their surrounding area. Nevertheless, location-dependent queries pose a serious privacy threat, as they may reveal sensitive information about an individual. To mitigate this threat, previous work has proposed the use of private information retrieval (PIR) protocols when querying the database. In this chapter, we argue that PIR queries are very expensive and should be avoided, to the extent possible. To this end, we propose a novel approach that allows mobile users to share anonymously their cached channel availability information that is obtained from previous queries. Our experiments with a real-life dataset, indicate that our methods reduce the number of PIR queries by 50% to 60%. Furthermore, they are efficient in terms of both computational and communication cost.

CHAPTER 4

PRIVACY-PRESERVING LOCATION-AWARE MOBILE ADVERTISEMENT

4.1 Background

Section 4.1.1 describes previous work on privacy-preserving targeted advertising, and Section 4.1.2 presents several privacy-preserving aggregation protocols.

4.1.1 Privacy-preserving targeted advertisement

The earliest work towards privacy-preserving advertisement is due to Juels [48]. He proposes the notion of a *negotiant*, which serves as a client-side proxy to protect user information and direct the targeting of advertisements. The idea is to allow each customer equipped with a public/private key to publish his ad request on a bulletin board. When enough ads are accumulated or some other triggering criterion occurs, a network of m non-colluding servers (advertisers) mixes the requests, and then uses distributed plaintext equality test to perform a blind lookup of consumer ad requests. At the final step, the servers apply quorum-controlled asymmetric proxy re-encryption to encrypt the ad with the corresponding customer's public key. The scheme by Juels does not consider other aspects of online advertising, such as privacy-preserving aggregation of ad statistics.

Adnostic [79] proposes a practical advertisement architecture that enables ad targeting without compromising user privacy. Specifically, behavioral profiling and ad targeting is privately done in the user's browser. The ad network periodically selects a number of ads that are sent (in plaintext) to the user's browser, where a rendering module chooses an ad for display. Adnostic's main goal is to complement existing behavioral advertising infrastructure, by providing efficient cryptographic billing based on homomorphic encryption and efficient zero-knowledge proofs (ZKPs). Nevertheless, their aggregation scheme makes use of a trusted third-party for decrypting the ad impression counters. The main drawback of Adnostic is that it leaks enough information which could allow the ad network to profile end users.

In a similar fashion to Adnostic, Privad [37] preserves privacy by maintaining profiles on the user's computer instead of the server. Furthermore, it employs an anonymizing proxy, called *dealer*, which sits between the clients and the ad network in order to hide any personally identifying information. The user first subscribes to advertisement categories of interest (called *channels*), by sending subscription requests to the ad network. The requests are encrypted with the ad network's public key and are, thus, indistinguishable to the dealer. In addition, every request includes a symmetric key that is used to encrypt the ads that are sent back to the client. Ad impressions are reported in a similar manner, i.e., encrypted with the ad network's public key. The main limitation of Privad is that it trusts that the ad network does not collude with the dealer.

MobiAd [38] proposes that end users keep a local private profile of categories of interest on their mobile devices. Ads are constantly broadcast on the local mobile base station, and the device's profile is responsible for downloading ads that are relevant to the user's interests. Statistics, such as CTRs or ad impressions, are updated via anonymization techniques. Specifically, MobiAd leverages the Delay Tolerant Networking paradigm, where every mobile user sends their statistics in a peer-to-peer store-and-forward method. To enhance privacy, dummy and camouflaging information is added to each forwarded statistic.

Finally, Hardt and Nath [40] provide a privacy-aware personalization scheme for mobile advertising. The main contribution of their work is the formalization of a common framework for personalized ad delivery, which can be instantiated at any required trade-off point between ad relevance, privacy, and efficiency. There are two major components in their approach. First, in an interactive manner, the client releases limited information to the ad network, such as a broad category of interest over a cloaked region. The server then pushes (in plaintext) the most relevant ads to the client, who filters them based on private criteria held at the device. The second component deals with the privacy-preserving and differentially private computation of ad impressions over a dynamic population. We discuss this component in the following section.

4.1.2 Privacy-preserving aggregation

One of the first application domains of privacy-preserving aggregation was wireless sensor networks. In this context, users are typically interested in computing aggregate information from sensor readings in a secure manner. To mitigate the limited resources of wireless sensor nodes, proposed methods employ *in-network* aggregation, i.e., sensors forward their encrypted readings to their parent aggregator until a designated *root* aggregator is reached. State-of-the-art protocols include [8] and [69], which are both based on efficient symmetric key encryption schemes that are additively homomorphic (they are simple variants of the one-time pad cipher). Nevertheless, such methods are not applicable in our problem setting, because they assume a trusted root aggregator that shares a secret encryption key with each sensor node, and can, therefore, decrypt all intermediate results.

To facilitate the design of efficient privacy-preserving aggregation protocols, numerous approaches in the literature leverage a *trusted third-party* to perform specific tasks [40, 47, 56, 75]. First, Shi et al. [75] assume a trusted key dealer that generates $s_0, s_1, \ldots, s_n \in \mathbb{Z}_p$, such that $\sum_{i=0}^{n} s_i = 0$. Prime p is the order of a cyclic group $\mathbb{G}$ with generator g for which Decisional Diffie-Hellman problem is hard. Here s_0 is the aggregator's share, whereas s_1 through s_n are distributed to the n users. At each aggregation period t, users compute ciphertext $c_i = g^{\hat{x}_i} \cdot H(t)^{s_i}$, where $\hat{x}_i$ is the noisy sample input of user i and H is a cryptographically secure hash function. The aggregator then computes $g^{\sum_{i=1}^{n} \hat{x}_i} = H(t)^{s_0} \cdot \prod_{i=1}^{n} c_i$, and solves the discrete log problem to recover the aggregate value of the noisy measurements. An extension of this scheme [11] incorporates fault tolerance for incoming or exiting users.

Jawurek and Kerschbaum [47] propose a fault-tolerant privacy-preserving scheme for computing statistics amongst smart meters. Their method involves three parties: (i) the data creators, such as smart meters, (ii) an untrusted service provider called data consumer, and (iii) a trusted key-managing authority. The aggregation part of the protocol is straightforward, and is built on the additively homomorphic Paillier cryptosystem. Specifically, the data creators use the key-managing authority's public key to encrypt their noisy readings, which are aggregated by the service provider. Lastly, the service provider relies on the key-managing authority for the decryption of the results. The authors also propose a protocol based on ZKPs that forces the service provider to prove to the key-managing authority that the aggregated measurements are fresh and have not been reused.

Hardt and Nath [40] introduce a differentially private aggregation protocol, as part of their privacy-aware mobile advertising scheme. It leverages a proxy server that is trusted not to collude with the ad network. In particular, each user i selects an encryption key k_i and sends (i) k_i to the ad network and (ii) $m_i = \hat{x}_i + k_i$ to the trusted proxy, where $\hat{x}_i$ is the noisy measurement. The proxy then forwards $\sum_{i=1}^{n} m_i$ to the ad network, which removes the encryption keys and retrieves the aggregate result.

Li et al. [56] assume a trusted key dealer that generates nc random secrets $s_1, \ldots, s_{nc}$. These secrets are divided into n random disjoint subsets, each containing exactly c secrets. The dealer then assigns to every user i one of the disjoint subsets $\mathcal{S}_i$, while providing the aggregator with $\mathcal{S} = \bigcup_{i=1}^{n} \mathcal{S}_i$. Knowledge of $\mathcal{S}$ allows the aggregator to decrypt the aggregate values, while staying oblivious to the individual inputs.

Although trusted third-parties simplify the construction of privacy-preserving protocols, they are not realistic for practical applications. To this end, Acs and Castelluccia [1], and Erkin and Tsudik [18] generate random secrets in a distributed manner among the participating users. Even though the underlying encryption protocols are different, both methods are very similar in the way they construct the secrets. Specifically, every user i maintains pairwise keys with every other user in the system. These keys are used to generate random values at each aggregation period, in order to mask the individual measurements. The random values are constructed in such a way that they collectively cancel out once they are aggregated, thus allowing the aggregator node to recover the results. Nevertheless, these methods are prone to collusions among the users. In particular, it is possible for $n-1$ users to reveal the keys they share with the n-th user, which in turn enables them to decrypt all measurements from that user (while maintaining their own input secret). On the other hand, our protocol is only vulnerable to full disclosure among the participating users.

Jung et al. [49] assume that the participating nodes are arranged in a circle, and each node i shares its public parameter with the two neighboring nodes. The public parameter is a value g^{r_i}, where g is a generator of a cyclic group $\mathbb{G}$ of prime order q (where the discrete log problem is hard) and r_i is selected uniformly at random from $\mathbb{Z}_q$. For the privacy-preserving `Sum` protocol, each node encrypts its input x_i as $C_i = (1+x_i \cdot p) \cdot R_i \bmod p^2$, where $R_i = (g^{r_{i+1}}/g^{r_{i-1}})^{r_i}$ and q divides $p-1$. The aggregator then computes $\prod_{i=1}^{n} C_i$ from which it retrieves $\sum_{i=1}^{n} x_i$. There are several drawbacks with this approach. First, it requires new keys at each aggregation period, which is very expensive. Second, it is vulnerable to collusion attacks that are very costly to defend against. Third, and most important, the encryption scheme employed by the authors is insecure and is easily broken with a simple brute-force attack. Note that, with an overwhelming probability, the measured value $(1 + x_i \cdot p)$ or any other uniformly random number in $\mathbb{Z}_{p^2}$ are not members of the cyclic group $\mathbb{G}$. Therefore,

C_i is also not a member of $\mathbb{G}$, which enables us to infer the value of R_i (and break the encryption) as follows: for all possible inputs x_i, compute $R = (1 + x_i \cdot p)^{-1} \cdot C_i \bmod p^2$; if $R^q = 1 \bmod p^2$, then $R_i = R$ and you found x_i.

Finally, Yang et al. [90] propose a scheme that is based on the additive homomorphism of the ElGamal cryptosystem. Specifically, each user i holds two pairs of keys $(x_i, X_i = g^{x_i})$ and $(y_i, Y_i = g^{y_i})$, such that g is generator of a group $\mathbb{G}$ where the discrete log problem is hard. The values x_i and y_i are private to each user, while X_i and Y_i are public. Let $X = \prod_{i=1}^{n} X_i$ and $Y = \prod_{i=1}^{n} Y_i$, which are known to all users. The aggregator needs to learn $d = \sum_{i=1}^{n} d_i$, where d_i is the individual contribution of user i. To this end, each user i sends to the aggregator $m_i = g^{d_i} \cdot X^{y_i}$ and $h_i = Y^{x_i}$. The aggregator computes $g^{\sum_{i=1}^{n} d_i} = \prod_{i=1}^{n} \frac{m_i}{h_i}$, and solves the discrete log problem to recover d. The main limitation of this scheme is that it requires expensive re-keying operations for every aggregation instance.

4.2 Preliminaries

Our methods are built on top of public key cryptosystems that are *additively* homomorphic. The additive homomorphism allows us to perform addition operations on plaintext values by manipulating their corresponding ciphertexts, i.e., without decrypting. For example, given two plaintext messages m_1 and m_2, and their respective ciphertexts $E(m_1)$ and $E(m_2)$, we can compute $E(m_1 + m_2) = E(m_1) \cdot E(m_2)$. Furthermore, we can multiply a message m with a plaintext value k as $E(k \cdot m) = E(m)^k$. Next, we describe briefly the three cryptosystems that we utilize in our protocols, namely the Paillier, BGN, and ElGamal cryptosystems.

4.2.1 Paillier cryptosystem

The Paillier cryptosystem [67] is an efficient protocol that can decrypt arbitrarily large plaintexts. It is semantically secure and its security is based on the Decisional Composite

Residuosity assumption. It consists of the following three functions.

Key generation. Choose two large primes q_1 and q_2 of equal length, and compute the RSA modulus $n = q_1 q_2$. The public key is n and the private key is $\varphi(n) = (q_1 - 1)(q_2 - 1)$.

Encryption. Given a message $m \in \mathbb{Z}_n$, choose a uniformly random value $r \in \mathbb{Z}_n^*$ and compute the ciphertext $c = (1+n)^m \cdot r^n \bmod n^2$.

Decryption. Given the ciphertext c, compute the plaintext $m = \frac{(c^{\varphi(n)} \bmod n^2) - 1}{n} \cdot \varphi(n)^{-1} \bmod n$.

4.2.2 Boneh, Goh, Nissim cryptosystem

The Boneh, Goh, Nissim (BGN) cryptosystem [5] makes use of finite groups of composite order that support bilinear maps. Let $\mathbb{G}$ and $\mathbb{G}_1$ be two cyclic groups of finite order $n = q_1 q_2$ and let g be a generator of $\mathbb{G}$. A map e is a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ if $\forall u, v \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}$, $e(u^a, v^b) = e(u, v)^{ab}$. It is also required that $e(g, g)$ is a generator of $\mathbb{G}_1$ for any choice of $g \in \mathbb{G}$. The scheme is semantically secure and its security is based on the subgroup indistinguishably problem, which is related to the hardness of computing discrete logarithms in the groups $\mathbb{G}$ and $\mathbb{G}_1$. The advantage of the homomorphic BGN cryptosystem is that it allows for *one multiplication* and arbitrary number of additions. On the other hand, decryption is based on a discrete log computation, so we are limited to small plaintext values. We use the following functions in our work.

Key generation. Choose two large primes q_1 and q_2 of equal length, and compute the RSA modulus $n = q_1 q_2$. Generate a bilinear group $\mathbb{G}$ of order n, and choose two random generators g and u. Let $h = u^{q_2}$ be a generator of a subgroup of $\mathbb{G}$ of order q_1. The private key is q_1, and the public key is h (the public key also includes the parameters n, $\mathbb{G}$, $\mathbb{G}_1$, e, and g).

Encryption in $\mathbb{G}$. Given a message $m \in \mathbb{Z}_{q_2}$, choose a uniformly random $r \in \mathbb{Z}_n$ and compute the ciphertext $c = g^m \cdot h^r \in \mathbb{G}$. Note that the system is additively homomorphic.

Multiplication. Given two ciphertexts $c_1 = g^{m_1} \cdot h^{r_1}$ and $c_2 = g^{m_2} \cdot h^{r_2}$, we can compute the encryption of $m_1 \cdot m_2 \bmod n$ as $c = e(c_1, c_2) = g_1^{m_1 m_2} \cdot h_1^{\hat{r}} \in \mathbb{G}_1$, where $g_1 = e(g, g)$ and $h_1 = e(g, h)$. Note that the system remains additively homomorphic in $\mathbb{G}_1$, but does not allow any further multiplications.

Decryption in $\mathbb{G}_1$. Given the ciphertext $c = g_1^m \cdot h_1^r$, compute $c^{q_1} = (g_1^{q_1})^m$ and solve the discrete log to retrieve m.

4.2.3 ElGamal cryptosystem

The ElGamal cryptosystem [67] is a protocol based on discrete logarithms. As such, it can only decrypt efficiently small plaintext values. It is semantically secure and its security is based on the Decisional Diffie-Hellman assumption. It consists of the following three functions.

Key generation. Let $\mathbb{G}$ be a cyclic group of prime order q, and let g be a generator of $\mathbb{G}$. Choose a private key x uniformly at random from $\mathbb{Z}_q$, and set the public key $h = g^x$.

Encryption. Given a message $m \in \mathbb{Z}_q$, choose a uniformly random $r \in \mathbb{Z}_q^*$ and compute the ciphertext $(c_1, c_2) = (g^r, h^{r+m})$.

Decryption. Given the ciphertext (c_1, c_2), compute $h^m = c_2 \cdot (c_1^x)^{-1}$ and solve the discrete log to retrieve m.

4.3 Privacy-preserving Ad Delivery

In this section, we present the details of our ad delivery protocol. Section 4.3.1 describes the system architecture and Section 4.3.2 presents the underlying threat model. Section

4.3.3 introduces our basic construction utilizing the Paillier cryptosystem and Section 4.3.4 presents the enhanced protocol that leverages the BGN cryptosystem.

4.3.1 System architecture

We consider a system where users travel around with their mobile devices and want to receive ads from businesses in their proximity. The ads can be requested manually by the user, or be periodically downloaded at the user's device and displayed inside popular smartphone apps. Ads are delivered through a dedicated ad network that bills the advertisers based on the number of times their ads are displayed to the users (ad impressions). To determine proximity, we assume that space is partitioned into a regular $N \times N$ grid, where each grid cell stores the ads that are physically located therein. Similarly, users map their GPS coordinates into a unique cell, and use the cell's position as an input to their query. Fig. 4.1 illustrates this approach, where a user located in Central Park (red pin) maps his location into cell $(3, 2)$.

Ads consist of a tuple $\langle id, cid, loc, text\rangle$, where id is the ad's unique identifier, cid identifies the advertisement category (e.g., food, shopping), loc contains the GPS coordinates of the ad's product, and $text$ is the message that is displayed to the user. We assume that the message is a few hundred bytes in size, and is padded (if needed) so that all ads have identical size. At the client side, we assume the existence of a profiling engine that monitors the user's browsing behavior and builds a private profile for that user. Alternatively, users may specify manually the types of ads that they want to receive, e.g., based on category, distance, keywords, etc. The exact profiling method is outside the scope of this paper.

Let $\mathcal{A}$ be the set of all ads, and $\mathcal{L}$ be the set of all locations (i.e., grid cells). (Table 4.1 contains a summary of symbols used in the remainder of this paper.) The set of ads $\mathcal{A}_l$ matching a location $l \in \mathcal{L}$ varies according to the location itself and the time of day. For example, some locations are more popular than others, while more ads may appear during

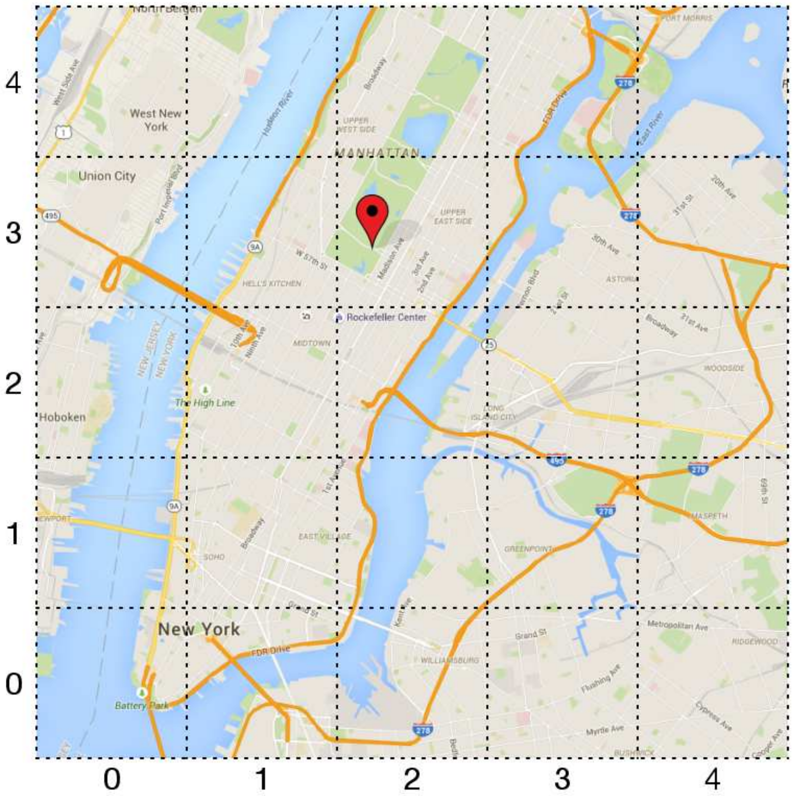


Figure 4.1: Example grid layout over the greater NYC area.

the day (such as lunch specials). We permit real time updates by the ad network, as long as the *id*s increase monotonically, such that no identifier is re-used until the completion of the current aggregation period.

4.3.2 Threat model

We consider the ad network (or any entity that has compromised the network's server) as the adversary, whose goal is to pinpoint a user's location into an area smaller than the whole data space. In this work, we aim at perfect privacy, i.e., the adversary should remain oblivious to the location of the users. We assume that the adversary is polynomial time and follows the *honest-but-curious* (or *semi-honest*) model, i.e., it will execute the protocol correctly, but will try to gain an advantage by examining the transcript of the messages that

Table 4.1: Summary of Symbols

Symbol	Description
n	Number of users in the system
N	Grid granularity
$\mathcal{A}$	The set of all advertisements
$\mathcal{L}$	The set of all locations
$\mathcal{A}_l \subset \mathcal{A}$	The set of ads matching location $l \in \mathcal{L}$
$\lvert\mathcal{A}_l\rvert_{max}$	Max number of ads across all locations $l \in \mathcal{L}$
$\mathcal{Q}, \mathcal{R}, \mathcal{C}$	Encrypted query vectors
$\mathcal{PK}$	Any public key
$\mathcal{P}$	Any private key
$\mathcal{B}$	Encrypted buffer
m	Number of ciphertexts required to store one ad

are exchanged during protocol execution.

4.3.3 Basic protocol

Our basic scheme is roughly based on the private stream searching protocol by Ostrovsky and Skeith III [66]. The intuition is to allow the client to specify the location of interest through an encrypted vector, and then use the additive homomorphism of the Paillier cryptosystem to encode the results into an encrypted buffer. We assume that all clients generate their Paillier keys locally when they first register in the system, and send the corresponding public keys to the server along with their queries. In what follows, we present our method in terms of three phases, namely, query generation, query processing, and result extraction.

Query generation. The client uses his public key $\mathcal{PK}$ to construct a vector $\mathcal{Q}$ of length $|\mathcal{L}| = N^2$, as shown in Algorithm 8. Every element in the vector is an encryption of 0, except for the element that corresponds to the user's location l, which is an encryption of 1. Due to the semantic security of the Paillier cryptosystem, these ciphertexts are indistinguishable to the adversary.

The query generation algorithm incurs an $\mathcal{O}(N^2)$ computational and communication cost,

Algorithm 8 Query generation (Paillier)

```
procedure GEN-QUERY-PAILLIER($\mathcal{PK}, loc$)
    map GPS location $loc$ into cell $l \in \mathcal{L}$;
    for each location $i \in \mathcal{L}$ do
        if $i == l$ then
            $\mathcal{Q}_i \leftarrow E_{\mathcal{PK}}(1)$;
        else
            $\mathcal{Q}_i \leftarrow E_{\mathcal{PK}}(0)$;
        end if
    end for
    return $\mathcal{Q}$;
end procedure
```

which is significant when N is large. However, one way to eliminate the online computational cost is to precompute offline (e.g., during night time, when the phone is charging) encryptions of 0, which is the major performance bottleneck at the client.

Query processing. After receiving the query vector $\mathcal{Q}$ and public key $\mathcal{PK}$, the ad network must process all the ads in $\mathcal{A}$ and return the relevant ones to the client. We assume that every ad fits in exactly m ciphertexts, where m depends on the underlying cryptosystem and key size. For example, if we use a 1024-bit RSA modulus (for Paillier), a 512-byte ad requires $m = 4$ ciphertexts. The main idea is to construct an encrypted buffer $\mathcal{B}$ that is capable of holding the maximum number of ads across any location $l \in \mathcal{L}$, i.e., $|\mathcal{A}_l|_{max}$. That is, the buffer should consist of exactly $m \cdot |\mathcal{A}_l|_{max}$ ciphertexts.

Algorithm 9 illustrates the procedure that generates the encrypted buffer at the ad network server. $\mathcal{B}$ is initially populated with encryptions of 0 (lines 2–3) that are computed with the client's public key (for efficiency, we use the same ciphertext for all entries). Next, the server processes the ads on a per cell basis. Each ad is split into m pieces, which are subsequently added into m consecutive locations on the buffer (lines 9–12). Note that, when $\mathcal{Q}_l$ is an encryption of 0, the process has no effect on the buffer contents. It is also worth noting that the round-robin manner in which we iterate over the buffer guarantees that there are no collisions when writing back the results. Consequently, when the procedure terminates, $\mathcal{B}$ contains (i) the encrypted ads corresponding to the queried location in successive order starting from a random position in the buffer, and (ii) encryptions of 0 at all the remaining positions.

Algorithm 9 Query processing (Paillier)

```
procedure GEN-BUFFER($\mathcal{PK}, \mathcal{Q}, \mathcal{A}$)
    for $i$ in 1 to $m \cdot |\mathcal{A}_l|_{max}$ do
        $\mathcal{B}_i \leftarrow E_{\mathcal{PK}}(0)$;
    end for
    $i \leftarrow 0$;
    for each cell $l \in \mathcal{L}$ do
        for each ad $a \in \mathcal{A}_l$ do
            split advertisement $a$ into $m$ pieces;
            for $k$ in 1 to $m$ do
                $\mathcal{B}_i \leftarrow \mathcal{B}_i \cdot {\mathcal{Q}_l}^{a_k}$;
                $i \leftarrow (++i)\%(m \cdot |\mathcal{A}_l|_{max})$;
            end for
        end for
    end for
    return $\mathcal{B}$;
end procedure
```

The computational cost at the server is $\mathcal{O}(m \cdot |\mathcal{A}|)$ modular exponentiations and multiplications, while the communication cost entails the transmission of $\mathcal{O}(m \cdot |\mathcal{A}_l|)$ ciphertexts. The Paillier cryptosystem is an ideal choice in this case, because it can decrypt arbitrarily large plaintexts, thus leading to an optimal value for m.

Result extraction. The result extraction procedure is fairly straightforward, i.e., the client simply decrypts with his private key $\mathcal{P}$ the ciphertexts that comprise buffer $\mathcal{B}$. Algorithm 10 summarizes the decryption process. Starting with the first ciphertext, the client decrypts it and checks whether it contains a useful ad. In particular, if the resulting plaintext M is 0, the corresponding position is empty. Furthermore, by decrypting the first part of the ad, the client recovers its *cid* value, thus determining (with the help of the profiling engine) whether the ad matches the user's profile. If the ad is not helpful to the client, the algorithm skips the remaining $m - 1$ ciphertexts and moves to the next ad (lines 4–6). Otherwise, the remaining ciphertexts are decrypted to reconstruct the entire ad (lines 8–12). When the algorithm terminates, the profiling engine moves the decrypted ads into the queue that is scheduled for display. The computational cost of this algorithm is $\mathcal{O}(m \cdot |\mathcal{A}_l|)$ decryption operations, which favors the Paillier cryptosystem, due its optimal m value.

Algorithm 10 Result extraction

```
 1: procedure GET-RESULTS($\mathcal{P}, \mathcal{B}$)
 2:     for $i$ in 1 to $m \cdot |\mathcal{A}_l|_{max}$ do
 3:         $M \leftarrow D_{\mathcal{P}}(\mathcal{B}_i)$;
 4:         if $M == 0$ or ad does not match interest then
 5:             $i \leftarrow i + m$;
 6:             continue;
 7:         else
 8:             initialize new ad with $M$;
 9:             for $j$ in 1 to $m - 1$ do
10:                 $M \leftarrow D_{\mathcal{P}}(\mathcal{B}_{i+j})$;
11:                 add $M$ to currently constructed ad;
12:             end for
13:             $i \leftarrow i + m$;
14:         end if
15:     end for
16:     return ads;
17: end procedure
```

Retrieving multiple cells. Some clients may desire to retrieve ads from multiple cells in their vicinity. Our protocol allows that, however, certain steps must be taken to ensure that there are no buffer collisions in the query processing phase. First, the encrypted buffer size should be set to $c \cdot m \cdot |\mathcal{A}_l|_{max}$ ciphertexts, where c is the number of cells marked by the client (the client should notify the server of the exact value of c). Second, the client should only mark cells that are processed by the server in a consecutive manner. Therefore, the two parties must agree on an arrangement of the cells that preserves geographical proximity. For this task, we can utilize the Hilbert space filling curve [50]. Hilbert space filling curve was described in Section 2.2.2. For example, a user located in cell 7 of Fig. 2.1 can request ads from cells 6–9 with a single query. In terms of performance, there is an increased communication cost in the query processing phase, and an increased computational cost at the client during result extraction. Nevertheless, all other costs remain unchanged.

Security. Note that the three phases of our basic scheme constitute a secure two-party computation protocol between the client and the server. As such, we can prove the security of the protocol for honest-but-curious adversaries, following the simulation paradigm [58]. It suffices to show that we are able to simulate the distribution of the messages that each party receives, given only the party's input and output in the protocol. The intuition is that, if we can simulate a party's messages knowing only their input and output, then the messages themselves cannot reveal any additional information.

First, the client's input consists of a binary query vector, and the output contains a number of ads matching a location. The only messages that the client receives from the server are a series of Paillier ciphertexts corresponding to $\mathcal{B}$. The simulator has knowledge of the client's public key and it also knows the decrypted ads. Therefore, it can simply reconstruct a version of the encrypted buffer from scratch. For the server, the input is the set of ads $\mathcal{A}$ and there is no output. The server only receives N^2 Paillier ciphertexts from the client, so the simulator can simply generate N^2 encryptions of 0. Given the semantic security of Paillier's cryptosystem, the server cannot distinguish these ciphertexts from the ones that are produced by the client's real input.

4.3.4 Query-efficient protocol

Our basic scheme is very efficient in terms of query processing and result extraction, but suffers from a $\mathcal{O}(N^2)$ cost in the query generation phase. As a result, it does not scale well for finer grids and is impractical for online queries, i.e., without offline pre-computations. To this end, we propose an enhanced version of the protocol that eliminates the need to send a unique ciphertext for every cell of the grid. The main idea is to identify the cell of interest with two encrypted binary vectors: one representing the rows and the other representing the columns. In Fig.4.1, for example, the plaintext row vector would be $(0, 0, 0, 1, 0)$ and the column vector would be $(0, 0, 1, 0, 0)$, thus pointing to the marked cell. The server would then need to multiply the corresponding bits for every cell, in order to determine whether the user is interested in that location or not (i.e., whether the result is 1 or 0). Next, we present the query generation and processing algorithms of this approach. Note that the result extraction phase is identical to the one in the basic protocol and is, thus, omitted.

Query generation. Clearly, our idea necessitates a cryptosystem that allows for both multiplication and addition of plaintexts in the ciphertext domain. This is the definition of fully homomorphic encryption [29] which, unfortunately, is not practical yet for real world applications. Fortunately, in our case, we only need to perform a single multiplication and an arbitrary number of additions, which is precisely what the BGN cryptosystem offers (Section 4.2.2). Therefore, as shown in Algorithm 11, the client uses his BGN public key $\mathcal{PK}$ to construct two encrypted vectors, $\mathcal{R}$ and $\mathcal{C}$, of length N. All the elements contain encryptions of 0, except for the ones that correspond to the user's row/column id. Due to the semantic security of the BGN cryptosystem, the ciphertexts are indistinguishable to an adversary.

Algorithm 11 is clearly more efficient than its Paillier counterpart, incurring a $\mathcal{O}(N)$ computational and communication cost at the client. Note that, similar to our basic protocol, users are also able to query for more than one location at a time, by setting multiple bits in the encrypted query vectors (assuming that the marked cells agree with the underlying Hilbert ordering).

Query processing. The query processing phase is identical to the one described in the

Algorithm 11 Query generation (BGN)

```
procedure Gen-Query-BGN($\mathcal{PK}, loc$)
    map GPS location $loc$ into cell $(i, j)$;
    for $k$ in 0 to $N - 1$ do
        if $k == i$ then
            $\mathcal{R}_k \leftarrow E_{\mathcal{PK}}(1)$;
        else
            $\mathcal{R}_k \leftarrow E_{\mathcal{PK}}(0)$;
        end if
        if $k == j$ then
            $\mathcal{C}_k \leftarrow E_{\mathcal{PK}}(1)$;
        else
            $\mathcal{C}_k \leftarrow E_{\mathcal{PK}}(0)$;
        end if
    end for
    return $\mathcal{R}, \mathcal{C}$;
end procedure
```

basic protocol, i.e., the server prepares an empty (encrypted) buffer $\mathcal{B}$ that eventually stores the ads that are relevant to the user's location. However, in this case, the server must first compute the query vector $\mathcal{Q}$ corresponding to all locations $l \in \mathcal{L}$. As shown in Algorithm 12, for each location $l \in \mathcal{L}$, the server computes $\mathcal{Q}_l = E_{\mathcal{PK}}(b_i \cdot b_j)$, where b_i and b_j are the query bits of the location's row and column ids. The ciphertext is computed through a bilinear map of the respective elements in $\mathcal{R}$ and $\mathcal{C}$, as explained in Section 4.2.2.

Algorithm 12 Query processing (BGN)

```
procedure Gen-Buffer-BGN($\mathcal{PK}, \mathcal{R}, \mathcal{C}, \mathcal{A}$)
    for each location $l \in \mathcal{L}$ do
        map $l$ into cell $(i, j)$;
        $\mathcal{Q}_l \leftarrow e(\mathcal{R}_i, \mathcal{C}_j)$;
    end for
    Gen-Buffer($\mathcal{PK}, \mathcal{Q}, \mathcal{A}$);
end procedure
```

The above algorithm entails $\mathcal{O}(N^2)$ bilinear map operations, as well as $\mathcal{O}(m \cdot |\mathcal{A}|)$ modular exponentiations and multiplications for constructing the encrypted buffer. On the other hand, the communication cost involves the transfer of $\mathcal{O}(m \cdot |\mathcal{A}_l|)$ ciphertexts. Compared to the basic scheme, we make the following two observations. First, the BGN-based protocol shifts the computational burden from the clients to the server. Instead of having the clients compute the query vector $\mathcal{Q}$, we provide the server with the minimal information needed to compute $\mathcal{Q}$ locally. This is significant improvement, because mobile devices have limited computational capabilities compared to a state-of-the-art many-core server.

Second, due to the discrete log nature of BGN, the value of m is significantly larger compared to the basic scheme. In our implementation, we choose to encrypt ads in 3-byte chunks, in order to take advantage of a pre-computed table of the first 2^{24} powers of $g_1^{q_1}$ (Section 4.2.2) and speed up the discrete log computation at the client. As a result, for a 512-byte ad, m is equal to 171, as opposed to 4 in the case of a 1024-bit Paillier key. Nevertheless, as we show in our experimental results, the overall cost is considerably lower compared to the basic scheme.

4.4 Privacy-preserving Collection of Ad Impressions

A fundamental component in a privacy-preserving ad network is its ability to compute aggregate statistics regarding the ads that are displayed to the users (ad impressions). In our system, we assume that the ad network collects the statistics at the end of each day, when most users' devices are idle and connected on the home WiFi network. The server first informs the clients of the total number of ads $|\mathcal{A}|$ that were scheduled that day and, for every ad in $\mathcal{A}$, each user must submit (privately) a bit indicating whether that ad was displayed or not. The server then aggregates the bits from all users, and updates the billing information for the underlying advertiser.

4.4.1 Threat model

We consider both the ad network and the clients as adversaries in this setting. They all follow the semi-honest adversarial model, and their goal is to identify any non-trivial information regarding individual measurements submitted by users. Note that, semi-honest behavior does not prohibit collusions among the different players, so users may collude by sharing private information. Our protocol has two desirable properties: (i) it does not employ a trusted third-party and (ii) it is secure against any number of colluding parties.

4.4.2 Aggregation protocol

Our aggregation protocol is based on a distributed version of the ElGamal cryptosystem, as given by Pedersen [71]. We chose the ElGamal cryptosystem, because of its simple key generation process and overall computational efficiency. The protocol consists of two phases, namely key generation and interactive aggregation, which are described next.

Key generation. All users and the server share a description of a cyclic group $\mathbb{G}$ of prime order q, and two generators of $\mathbb{G}$, namely g and y (Section 4.2.3). The objective is for the n users to collectively compute a public key $h = g^x$, such that $x = \sum_{i=1}^{n} x_i$ is the private key and x_i is user i's secret share of the key. In other words, the private key is distributed to all users in the system and, therefore, decryption necessitates input from all n users.

Algorithm 13 illustrates the key generation process. Initially, each user selects a random secret $x_i \in \mathbb{Z}_q$ and commits [70] to input g^{x_i}, by sending a commitment $C_i(g^{x_i}, r_i)$ to the server. The commitment is simply an encryption of the user's input with a random key r_i (line 4), and its purpose is to prohibit users from modifying their inputs in the later stages of the algorithm. After a user downloads the set of all commitments, he reveals his public input by sending the tuple (g^{x_i}, r_i) to the server (lines 7–8). Finally, each user verifies all commitments and computes locally the ElGamal public key h (lines 11–13). In terms of performance, the algorithm requires $\mathcal{O}(n)$ modular exponentiations and multiplications, and involves the exchange of $\mathcal{O}(n)$ ciphertexts and random secrets.

Algorithm 13 Distributed key generation

```
procedure Gen-Key($\mathbb{G}, q, g, y$)
    for each user $i$ do
        select $x_i$ and $r_i$ uniformly at random from $\mathbb{Z}_q$;
        upload commitment $C_i(g^{x_i}, r_i) = g^{x_i} \cdot y^{r_i}$;
    end for
    for each user $i$ do
        download commitments $C_1, \ldots, C_n$;
        upload $(g^{x_i}, r_i)$;
    end for
    for each user $i$ do
        download $(g^{x_i}, r_i)$ for $i \in \{1, \ldots, n\}$;
        verify $C_i(g^{x_i}, r_i) = g^{x_i} \cdot y^{r_i}$ for $i \in \{1, \ldots, n\}$;
        compute public key $h = \prod_{i=1}^{n} g^{x_i}$;
    end for
end procedure
```

An important feature in a privacy-preserving aggregation protocol is efficient key management. Prior methods that do not employ trusted third-parties, such as [49] and [90], require expensive re-keying operations for each aggregated value. On the other hand, our approach leverages the ElGamal cryptosystem with a pre-established key, so all values are aggregated under the same public key. Furthermore, our method handles user deletions trivially. In particular, when user i leaves the system, the public key is updated as $h' = h \cdot (g^{x_i})^{-1}$, i.e., the user's secret share is removed from the private/public key. New users, however, necessitate the invocation of the distributed key generation algorithm. To reduce the cost of frequent key generation operations, the server may choose to perform batch insertions.

Interactive aggregation. Having established the public encryption key h, the aggregation of the ad impressions is performed at the ad network's server. Algorithm 14 summarizes the aggregation protocol for a single ad. The first step is for all users to upload their encrypted bits b_i at the server (lines 3–4). Next, the server leverages the additive homomorphism of

ElGamal to produce the encryption of $b = \sum_{i=1}^{n} b_i$ (lines 6–7). However, the server is unable to decrypt the result, and has to rely on the n users to perform the decryption. Observe that the decryption function necessitates the computation of h^{-r}, which is equal to $(g^r)^{-x}$. To this end, each user downloads g^r and submits to the server $(g^r)^{-x_i}$ (lines 9–10). Finally, the server aggregates these values to compute h^{-r} and proceeds to recover b (lines 12–14). The overall (across all ads) computational cost at the client consists of $\mathcal{O}(|\mathcal{A}|)$ modular exponentiations, while the communication cost entails the exchange of $\mathcal{O}(|\mathcal{A}|)$ ciphertexts.

Algorithm 14 Aggregation protocol for a single value

```
 1: procedure GEN-AGGREGATE($\mathbb{G}, q, g, h$)
 2:     for each user $i$ do
 3:         select $r_i$ uniformly at random from $\mathbb{Z}_q$;
 4:         upload encrypted bit $b_i$ as tuple $(g^{r_i}, h^{b_i + r_i})$;
 5:     end for
 6:     server: compute $g^r = \prod_{i=1}^{n} g^{r_i}$;
 7:     server: compute $h^{b+r} = \prod_{i=1}^{n} h^{b_i + r_i}$;
 8:     for each user $i$ do
 9:         download $g^r$;
10:         upload $(g^r)^{-x_i}$;
11:     end for
12:     server: compute $h^{-r} = \prod_{i=1}^{n} (g^r)^{-x_i}$;
13:     server: compute $h^b = h^{-r} \cdot h^{b+r}$;
14:     server: solve discrete log to recover $b = \sum_{i=1}^{n} b_i$;
15:     return $m$;
16: end procedure
```

Security. Our aggregation protocol inherits the semantic security of the underlying ElGamal cryptosystem. Furthermore, the distributed implementation guarantees that no group of less than n users is able to decrypt a submitted bit. As such, to compromise the privacy of a single user, all other $n - 1$ users have to reveal their own bits, which is not a weakness of the protocol itself. Note that, to maintain privacy across dynamic populations, we may easily apply the concept of differential privacy [1], by having each user submit noisy measurements. Nevertheless such methods are orthogonal to this work.

4.5 Experimental Evaluation

In this section we present the results of our experimental evaluation. Section 4.5.1 describes the setup of the experiments and Section 4.5.2 presents our results.

4.5.1 Setup

We implemented the three homomorphic cryptosystems from Section 4.2 with the C programming language, using the GMP[1] multiple precision arithmetic library. For the BGN cryptosystem, we leveraged Ben Lynn's PBC library[2] that is also written on top of GMP. We tested the protocols on two different architectures, namely a 3.5 GHz Intel i7 CPU (x86_64) representing the server, and an Apple A8 CPU (arm64) representing the client. Note that, due to some porting problems with GMP's source code, we were unable to compile GMP with the assembly optimizations for the arm64 device. As a result, the client CPU times may be underestimated.

For security, we chose a 1024-bit RSA modulus for the Paillier and BGN cryptosystems, and a 160-bit ElGamal key. Table 4.2 summarizes the computational cost of the basic cryptographic primitives at the two different architectures. Missing values imply that the underlying operation is not required. The resulting ciphertext sizes are 256 bytes for Paillier and ElGamal, and 260 bytes for BGN.

In the following section, we measure the computational and communication costs of the various components of our methods, at both the client and server. Table 4.3 lists the parameters that control our experiments. In each experiment we vary a single parameter and keep the remaining ones to their default values. Note that, given the unique properties of our mobile advertisement framework, we do not compare against other, more computationally

[1] `https://gmplib.org/`

[2] `https://crypto.stanford.edu/pbc/`

Table 4.2: Cost of cryptographic primitives at client and server (all times in ms)

Crypto primitive	**Server**	**Client**
Paillier encryption	–	15.7
Paillier decryption	–	15.7
Paillier exponentiation (128-byte exponent)	1.2	–
Paillier multiplication	0.002	–
BGN encryption	–	9.8
BGN decryption	–	7.2
BGN exponentiation (3-byte exponent)	0.013	–
BGN multiplication	0.002	–
BGN bilinear map	5.5	–
ElGamal encryption	–	1.4
ElGamal exponentiation (20-byte exponent)	–	0.7
ElGamal multiplication	0.001	0.01

efficient approaches, because they do not provide the same level of security.

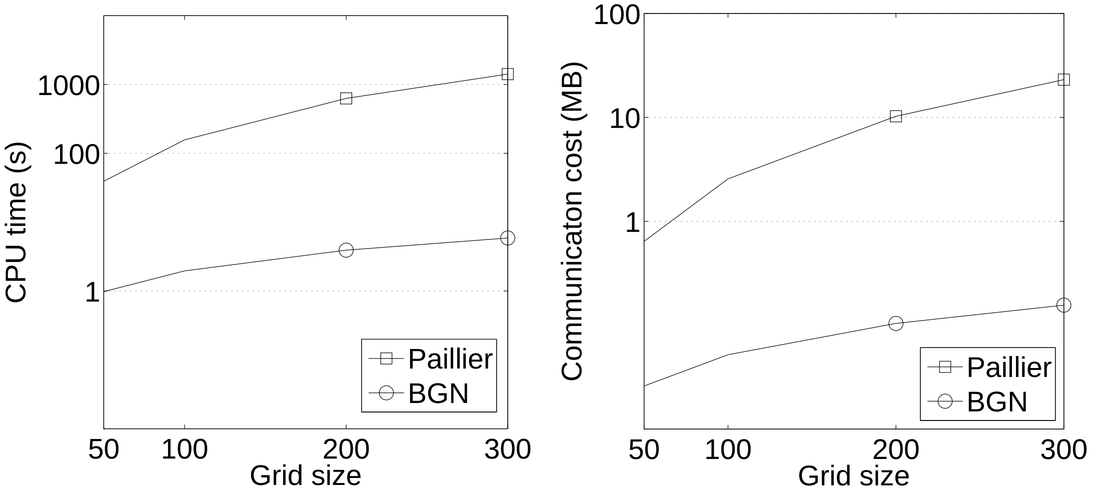
Table 4.3: Experimental parameters

Parameter	**Values**	**Default**
Grid granularity (N)	50, 100, 200, 300	100
Number of ads ($\|\mathcal{A}\|$)	1K, 2K, 5K, 10K	2K
Number of users (n)	10K, 20K, 50K, 100K	20K
Max ads per location ($\|\mathcal{A}_l\|_{max}$)	20, 50, 100, 200	50
Ad size (bytes)	256, 512, 1024, 2048	512

4.5.2 Results

In the first experiment, we investigate the performance of the client's query generation algorithm for the two ad retrieval protocols. Fig. 4.2a shows the CPU time at the client (logarithmic scale) as a function of the grid granularity.

For the default 100×100 grid, the Paillier-based scheme takes over 150 seconds of compute time, whereas BGN terminates in just 2 seconds. For finer grids, the cost of the basic protocol becomes prohibitive. This is due to the quadratic complexity of Algorithm 8 that generates one ciphertext for each of the N^2 cells. On the other hand, the BGN-based protocol is



(a) (b)

Figure 4.2: Query generation cost at the client vs. grid size (a) CPU time (a) Communication cost

very efficient, requiring less than 7 seconds of compute time even for the 300 × 300 grid. Fig. 4.2b illustrates the communication cost for the same experiment. Paillier's quadratic cost is again evident, as it necessitates 0.6–23 MB of data transfer per query. Alternatively, the BGN-based scheme incurs less than 160 KB of communication cost under all settings.

Staying at the client side, we measure the cost of the result extraction procedure (i.e., buffer decryption) as a function of the buffer size ($|\mathcal{A}_l|_{max}$). We assume that approximately 20% of the total ads in each location will match the client's profile. Therefore, approximately 80% of the ads entail a single decryption operation, while the rest invoke all m decryptions (as explained in Algorithm 10). Fig. 4.3a demonstrates the computational efficiency of the basic scheme, which is about 12 seconds faster than BGN for all buffer sizes.

This is due to the discrete log nature of BGN that necessitates numerous ciphertexts to encrypt a single ad. In particular, for our default settings, $m = 171$ for BGN and $m = 4$ for Paillier. Even though the decryption operation is twice as fast with BGN (Table 4.2), the sheer amount of operations needed negate this advantage. Fig. 4.3b shows the communication cost for downloading the encrypted buffer $\mathcal{B}$ from the server. Both methods scale linearly

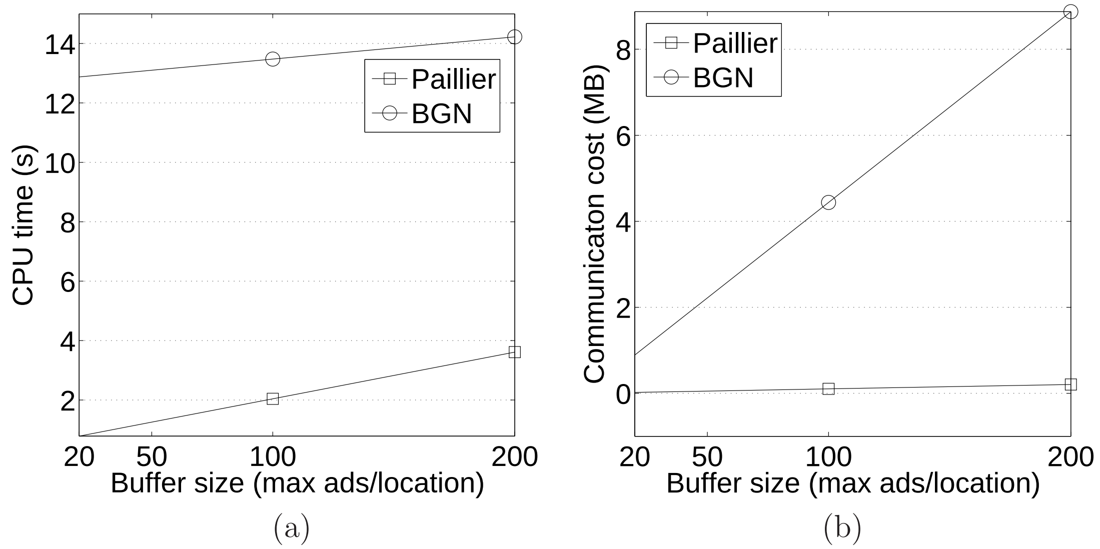


Figure 4.3: Result extraction cost at the client vs. buffer size (max ads/location) (a) CPU time (b) Communication cost

with the buffer size, but the basic protocol is clearly superior, due to its optimal m value.

In the next experiment, we investigate the performance of the result extraction algorithm as a function of the ad size. As shown in Fig. 4.4a, the decryption time grows linearly with the ad size, however, the Paillier-based curve grows at a much slower rate.

Again, this is due to the lower value of m that necessitates just 16 ciphertexts for a 2048-byte ad, as opposed to 683 for BGN. The same pattern is observed in Fig. 4.4b, where we measure the communication cost of this experiment. BGN involves the transfer of a few MB of data from the server, while Paillier's cost remains under 210 KB. Nevertheless, in a real world application, we expect the ads to be just a few hundred bytes in size. They would consist of plain text that is easily compressed to improve efficiency. In this setting, the BGN-based approach can decrypt ads in real time.

To get a concrete picture of the relative performance of the two ad delivery protocols, Fig. 4.5 shows, in logarithmic scale, the cumulative cost (query generation plus result extraction) at the client as a function of the grid size.

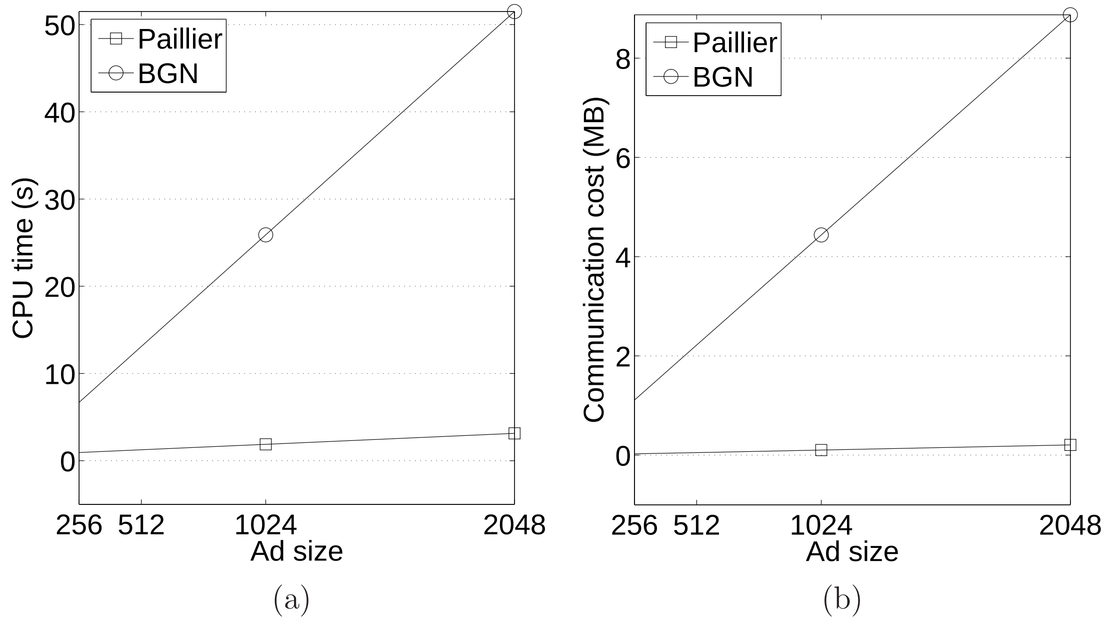
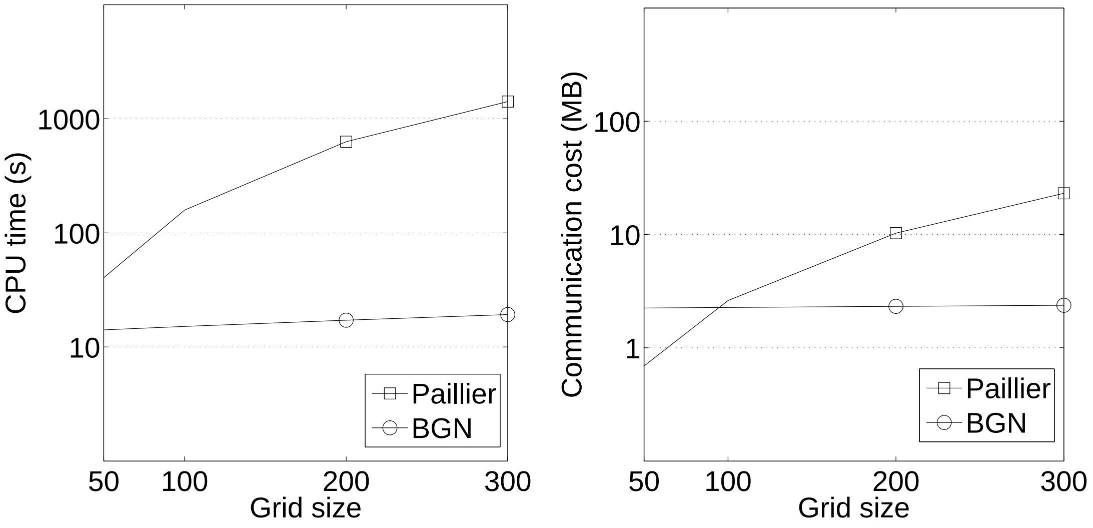


Figure 4.4: Result extraction cost at the client vs. ad size (a) CPU time (b) Communication cost

In terms of CPU time (Fig. 4.5a), the BGN protocol is the clear winner in all settings. Even for a coarse 50×50 grid, BGN is three times faster than Paillier (41 vs. 14 seconds), with that gap growing fast as N increases. The only advantage of the basic scheme is in the cumulative communication cost (Fig. 4.5b), where it slightly outperforms BGN for coarse grids. Note that the communication cost of BGN remains almost constant at around 2.2 MB, i.e., the cost of downloading the encrypted buffer.

Even though the basic scheme is clearly outperformed by its BGN counterpart, it could still be very useful in certain situations, given its efficiency in the result extraction phase. As explained in Section 4.3.3, the bottleneck of the Paillier query generation algorithm is the computation of N^2 ciphertexts. However, these ciphertexts consist almost entirely of encryptions of 0, and are independent of the client's location. Therefore, it is not inconceivable to imagine a scenario where the client pre-computes offline a large pool of ciphertexts that can be used in future queries. These pre-computations could be performed at night time, when



(a) (b)

Figure 4.5: Cumulative cost at the client (query generation + result extraction) vs. grid size (a) CPU time (b) Communication cost

the user's device is idle and possibly charging. In this case, the user may select in real time the cryptosystem that he wants to employ. For example, if a sufficient number of ciphertexts exists in the pool, choose the Paillier cryptosystem; otherwise, the only reasonable choice is BGN.

We next shift our focus towards the server, and investigate its performance in the query processing phase of our protocol. First, Fig. 4.6a illustrates the query processing time at the server as a function of the total number of ads $|\mathcal{A}|$.

As expected, both methods scale linearly with $|\mathcal{A}|$, since the server must perform m modular exponentiations and multiplications for every ad in the system. Recall that the intuition behind the BGN protocol was to shift the computational burden from the mobile devices to the server. As such, the server must perform N^2 bilinear map computations before processing the actual ads (Algorithm 12). As shown in Table 4.2, this is by far the most expensive cryptographic operation at the server and, for the 100×100 grid, this preprocessing alone takes 55 seconds. Nevertheless, the actual processing of the ads is faster with BGN,

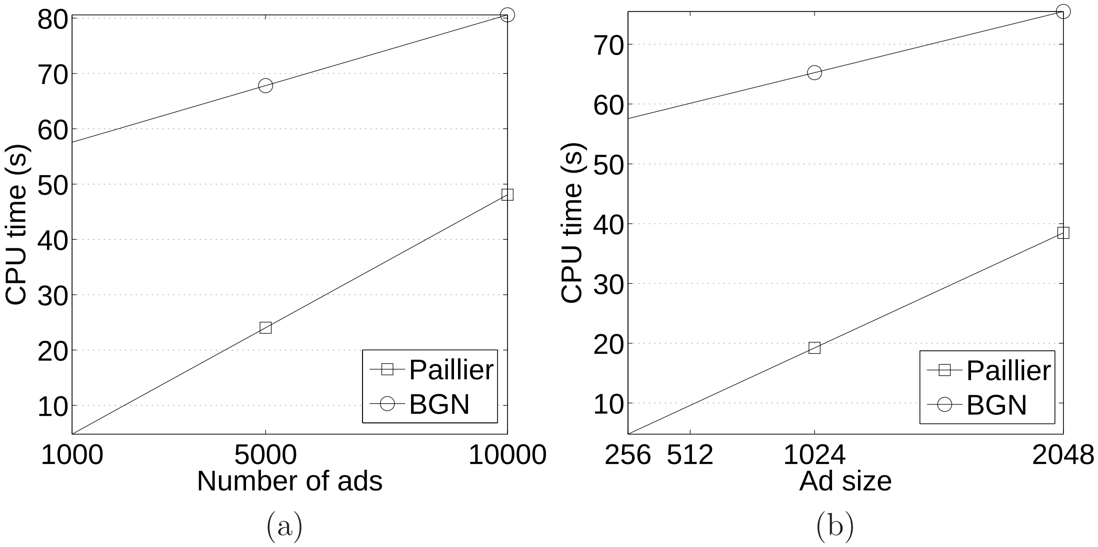


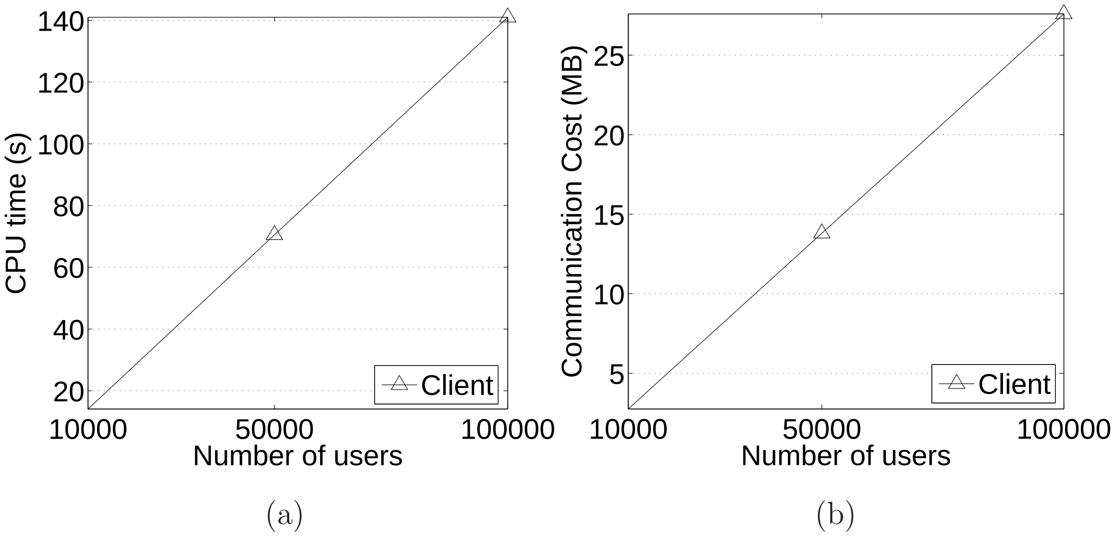
Figure 4.6: Query processing cost at the server (a) CPU time vs. number of ads (b) CPU time vs. ad size

so the performance gap against Paillier closes when $|\mathcal{A}|$ increases.

Similarly, Fig. 4.6b shows the query processing time at the server as a function of the ad size. The trend observed is similar to Fig. 4.6a, because larger ads necessitate more computations, i.e., the cost is linear in m. Even though these processing times appear significant, they correspond to a single CPU with serialized execution. State-of-the-art servers today consist of possibly tens or even hundreds of compute units that, in our case, could be used to process tens of ads in parallel.

In the last set of experiments, we measure the costs associated with the privacy-preserving collection of ad impressions. We start by examining the cost of the distributed key generation algorithm. Fig. 4.7a illustrates the CPU time spent at the client as a function of the total number of users n.

This cost is dominated by the modular exponentiations that are required to verify the commitments submitted by the remaining $n-1$ users. As such, the cost grows linearly with n, ranging from 14 to 141 seconds. This is also true for the communication cost (Fig. 4.7b) that consists of the cost of downloading the users' commitments and public key shares (Algorithm



(a)

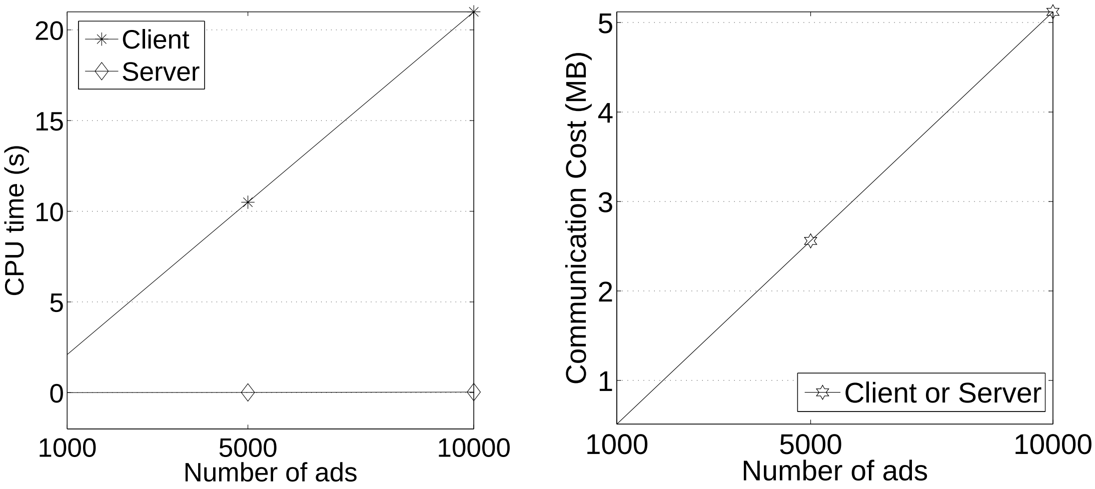
(b)

Figure 4.7: Key generation cost at the client vs. number of clients in the system (a) CPU time (b) Communication cost

13). The communication cost is moderate, ranging from 2.7 to 27 MB.

There are two observations to be made here. First, the key generation algorithm is not invoked frequently (only when new users enter the system), and the ad network has the option to delay this process in order to perform batch insertions. In addition, key generation can be performed at night, when the clients' devices are idle. The second observation is that the ad network could split users into multiple groups, in order to speed up the key generation process. That is, every group of users (say 10,000) would generate their own public key to use in the aggregation algorithm.

Finally, Fig. 4.8 shows the cost of the interactive aggregation process at both the client and server, as a function of the total number of ads $|\mathcal{A}|$. Clearly, both the CPU and communication costs are linear in $|\mathcal{A}|$, because the client has to submit one ciphertext (ad impression) for every ad in the system, regardless of whether that ad was displayed or not. In addition, the client is involved in the decryption process of $|\mathcal{A}|$ ciphertexts that contain the aggregated ad impressions by all users. The CPU time at the client (Fig. 4.8a) is dominated by the $|\mathcal{A}|$ encryption operations, each costing 1.4 ms. On the other hand, the decryption process necessitates just $|\mathcal{A}|$ modular exponentiations. At the server side, the computational cost per client is very low, as it entails $2 \cdot |\mathcal{A}|$ modular multiplications (with just 1 μs per operation). The communication cost (Fig. 4.8b) for the two parties is low, ranging from 0.5–5 MB. It consists of the interactive exchange of ciphertexts between the client and the server (Algorithm 14).



(a) (b)

Figure 4.8: Aggregation cost vs. number of ads (a) CPU time (b) Communication cost

4.6 Summary

In this chapter we present the first location-aware mobile advertising framework that offers stringent privacy guarantees through cryptographic constructions. Unlike previous work, the methods presented in this chapter guarantee that the ad network is oblivious to both the content sent to the clients, and the ad impressions submitted by the clients. Furthermore, we do not employ trusted third-parties and our protocols are secure against any number of colluding parties. We implemented the underlying cryptographic primitives on mobile devices and showed that our framework is practical for real world applications. Experiments based on location-aware mobile devices show that our novel methods are both computational and communicational efficient

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

DSA allows license-exempt users to access licensed spectrum band when not in use by their respective owners. More specifically, in the database-driven DSA model, mobile users issue location-based queries to a white-space database, in order to identify idle channels in their area. To preserve location privacy, existing solutions suggest the use of private information retrieval (PIR) protocols when querying the database. However, the proposed state-of-the-art solutions are not communication efficient and fail to take into account user mobility. To this end, our work addresses these shortcoming by introducing an efficient privacy-preserving protocol which leverages the Hilbert space filling curve and the communication efficient PIR scheme of Gentry and Ramzan. Our work provides optimizations for mobile users that require privacy on-the-fly and for users that have full a priori knowledge of their trajectories. Experimental results, based on two real life datasets, show that compared to the current state-of-the-art protocol, our work reduces the query response time at the mobile clients by a large factor. Nonetheless, PIR protocols are very expensive and may lead to significant costs for highly mobile clients. Therefore, in the second part of our work towards location privacy in the context of database-driven DSA, we introduce a novel method which allows wireless users to collaborate in a peer-to-peer (P2P) manner and share their cached channel availability knowledge in a privacy-preserving method. Here we leverage an anonymous veto protocol which anonymizes the exchange of information among a group of users. Using measurements based on a real life dataset, we show that our P2P protocol reduces the number of PIR queries by 50% to 60%, while incurring low computational and communicational costs.

Location-aware mobile advertising is an ever expanding field which is forecast to grow much faster than any other industry in the digital area. There exists a clear conflict between location-privacy and location-aware mobile advertisement, yet existing advertising practices rely heavily on non-disclosure agreements and policy enforcements rather than computational privacy guarantees. To this end, our work is the first to introduce a novel privacy-preserving location-aware mobile advertisement framework which is build with privacy in mind from the ground up. Our work guarantees, through cryptographic constructions based on the Paillier and Boneh, Goh, Nissim cryptosystems, that mobile users can receive advertisements relative to their location and interests in a privacy-preserving manner. Furthermore, through cryptographic constructions based on a distributed ElGamal cryptosystem, our work enables the advertisement network to compute aggregate statistics of ad impressions and click-through-rates in a privacy-preserving manner. Through extensive experimentations, we show that the methods introduced in this part of our work are efficient in terms of both computational and communicational costs, especially at the client side.

5.2 Future Work

Dynamic spectrum access has been proposed as a viable framework for maximizing the usability of the wireless spectrum, by allowing some portions of it to be accessed and used in a dynamic manner. Until now, the FCC has allowed only TV bands ranging from 512-608 Mhz (TV channels 21-36) and 614-698 Mhz (TV channels 38-51) to be accessed dynamically. The de facto method of performing spectrum sensing for any mobile device that wishes to utilize TV bands in a DSA manner is by querying a white-space database. Specifics of this method were explained in Section 1.

However, with the exponential rise of the connected mobile devices it is predicted that more spectrum might have to be designated as dynamically accessible. The future direction is exemplified from initiatives such as the National Broadband Plan (NBP) whose goal is to identify 300 Mhz of spectrum, in the course of the next 5 years, that can be utilized for broadband use. The National Oceanic and Atmospheric Administration (NOAA) frequency bands between 1675-1710 Mhz have been designated as dynamically accessible. The NOAA frequency bands are primarily used by weather balloons and weather satellite down links as well as from universities and private weather forecasting entities. Therefore, unlike the white-space TV bands, which are reserved for scheduled programming days ahead of time, the sudden and unexpected usage, by its primary incumbents, make NOAA allocated bands unfit for the database-driven dynamic access model. As such a more advanced spectrum sensing model would have to preside in order to accommodate dynamic utilization of such randomly occupied spectrum bands in light of their unpredictability.

One computing model that has recently attracted much attention in literature is the participatory crowdsourcing model. This model leverages the sensing, processing, storing and reporting power of the crowds, equipped with commodity hardware, towards the computation of a specific function over a geographical area. Examples of participatory crowdsourcing range from traffic monitoring to air and noise quality control to monitoring and modeling of the spread of infectious diseases [14]. More importantly, recent literature studies such as [64] and [94] have investigated the feasibility of spectrum monitoring through cheap commodity hardware by leveraging participatory crowdsourcing. Results from these studies indicate that real time spectrum sensing, monitoring and reporting by means of participatory crowdsourcing equipped with commodity hardware is indeed feasible in the very near future.

In the most basic model, the main parties involved in participatory crowdsourcing are (i) a set of users (crowd) and (ii) a crowdsourcing server. The two parties interact with each other through the following processes. The crowdsourcing server accepts queries from the set of users. The very first step towards any participatory crowdsourcing is the *task assignment* or simply *tasking*. Here, the participatory crowdsourcing server attempts to determine a subset of the set of users users which fit the criteria specified in the query and who are willing to participate towards the specified task. Consequently the server outsources the task to the participating users.

In the example of real time spectrum sensing, mobile users interested into accessing the NOAA licensed bands in a DSA manner, issue spectrum report queries to the crowdsourcing server. Each query contains the latitude/longitude coordinates of a minimum bounding region (MBR) for which the querying user wants to learn spectrum availability. We assume that there exist no peer to peer communication links between the users in the system, i.e. spectrum report queries can only be sent to the crowdsourcing server, which replies with a spectrum report. In this context, the *tasking* function would identify participating mobile users, which are geographically located within the boundaries of the tasking region and who can perform spectrum scanning via their commodity hardware. Next, any participating user, would utilize his own *sensing* capabilities to collect spectrum band readings as specified by the assigned task. A local processing and storage step may be required before each participating user *reports* his individual sensing report to the crowdsourcing server. The crowdsourcing server *fuses* the reported spectrum readings and *presents* the overall compiled results to the querying user.

It is evident, by considering the basic model explained in the previous paragraph, that participatory crowdsourcing can lead to severe privacy issues. By examining the sequence of processes involved in the most basic model of participatory crowdystem, one can immediately notice that location privacy is at risk. First and foremost, the query containing the MBR coordinates needs to be protected, i.e. two consecutive queries for the same MBR should be *computationally indistinguishable* at the server. Furthermore, since participating users will report their sensor readings based on location dependent tasks their location should also be preserved.

Secondarily, the nature of the sensor readings, even if stripped of any location specific information, can lead to location disclosure, assuming that the attacker has background information regarding the system settings. Correlation between the number of free channels in the NOAA licensed bands and physical locations is possible. For example, reports identifying very low number of free channels in the NOAA licensed bands, can be used to correlate the reports to locations which are near meteorological stations.

Currently there exist no proposed method in literature which considers the problem of dynamic spectrum sensing and reporting through participatory sensing model. Vu et al. [88] consider the more generic problem of location privacy in participatory sensing. Their proposed work revolves around the concept of *k-anonymity* and their scheme makes use of locality-sensitive hashing in order to partition reporting user locations in groups which contain at least k users. Their proposed scheme offers does not protect the initial query issued to the crowdsourcing system.

Shi et al. [76] consider the problem of data aggregation in people-centric urban sensing systems. They propose PriSense, a method based on the concept of data slicing and mixing which can support a wide range of statistical additive and non-additive aggregation functions. The basic idea is for each node to slice its data into slices which are randomly distributed amongst other nodes. Each node sends to the aggregation server the sum of its own slice and the slices received from other nodes in order to calculate the final function. In our model, it is infeasible for nodes that are dispersed over a geographical location to communicate in a peer to peer manner. Furthermore, PriSense crowdsourcing model omits the initial *tasking* that is done from the server. It simply assumes that nodes initiate a sensing session which needs to be aggregated at the server. Lastly, only the data privacy is considered i.e., node's individual reports are aggregated in a privacy-preserving manner. Location privacy is not considered in PriSense.

Qiu et al. [73] propose SLICER, a slicing-based k-anonymous privacy preserving scheme for participatory sensing. Similar to the scheme presented by Shi et al. [76], SLICER incorporates erasure coding to encode each sensing record into a number of small slices which are then transferred to other participants. The participants report the slices to the service provider, which decodes the original record captured during the sensing period. Unfortunately, the SLICER method requires that the generator of the slice as well as the participants communicate with each other in order to attain *k-anonymity*. Furthermore, SLICER provides no privacy preservation of any queries directed to the crowdsourcing server.

Gao et al. [26] proposed TrPF, a trajectory privacy-preserving framework for participatory sensing which makes use of mix network for the anonymization of sensing reports, and trusted third party, which stores participators relevant information such as certifications and pseudonyms. The trusted third party assumption is not realistic even though constructions based on trusted third party simply the overall construction design.

Chakraborty et al. [10] propose a framework for context-aware privacy of sensor data on mobile systems. Unlike other models which rely on anonymization techniques in order to hide the identity of the users, they take the more general approach of choosing what data to share, in such a way that certain kinds of sensitive inferences cannot be drawn. The authors propose *ipShield* which uses current user context together with a user's behavior in order to quantify an adversary's knowledge regarding a sensitive inference, however obfuscation mechanisms are outside the scope of the proposed method by Chakraborty et al.

Li et al. [56] make use of additive homomorphic encryption combined with a new key management scheme to reduce the communication and encryption overhead. In particular they rely on a trusted key dealer to generate nc random secrets $s_1, ..., s_{nc}$ which are divided into n random disjoint subsets containing c secrets. The dealer assigns to every user one of the disjoint subsets while providing the aggregator with the union of all disjoint subsets.

The authors extend their idea by a novel ring-based interleaved grouping technique that diminishes the number of participants that need to renew their cryptographic keys. This scheme does not consider protection of the incoming queries directed at the crowdsourcing server and in particular does not consider location privacy of the reporting sensors.

As future work we would like to tackle the problem of privacy-preserving participatory crowdsourcing in the context of DSA. In particular we would like to provide a comprehensive cryptographic construction which can protect the privacy of (i) incoming queries that are being issued to the crowdsourcing server (ii) the location of the participating users (iii) the measurements reported by the participating users (iv) the fused spectrum sensing report contained in the server reply.

5.3 Complete List Of Publications Under This Dissertation

The following is a cumulative list of all (authored/co-authored) published work and/or work submitted for publication which was developed during the course of the dissertation.

1. Erald Troja, Kenneth Ezirim, Shamik Sengupta, and Michael Hannon. Performance evaluation of RODEO: Route degradation optimization for the multi-hop dynamic spectrum access networks. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 232–236. IEEE, 2013 [85].

2. Kenneth Ezirim, Shamik Sengupta, and Erald Troja. (multiple) channel acquisition and contention handling mechanisms for dynamic spectrum access in a distributed system of cognitive radio networks. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 252–256. IEEE, 2013 [19].

3. Erald Troja, Kenneth Ezirim, and Suman Bhunia. Route aware dynamic channel scheduling and selection for multi-hop cognitive radio networks. In *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, pages 1–5. IEEE, 2013 [84].

4. Kenneth Ezirim, Erald Troja, and Shamik Sengupta. Sustenance against rl-based sybil attacks in cognitive radio networks using dynamic reputation system. In *Military Communications Conference, MILCOM 2013-2013 IEEE*, pages 1789–1794. IEEE, 2013 [20].

5. Erald Troja and Spiridon Bakiras. Efficient location privacy for moving clients in database-driven dynamic spectrum access. In *Proceedings of the 24th International Conference on Computer Communications and Networks*. IEEE, 2015 [82].

6. Erald Troja and Spiridon Bakiras. Leveraging P2P interactions for efficient location privacy in database-driven dynamic spectrum access. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 453–456. ACM, 2014 [81].

7. Erald Troja and Spiridon Bakiras. Leveraging P2P interactions for efficient location privacy in database-driven dynamic spectrum access. *International Journal of Network Security*, 17(5):569–579, 2015 [83].

8. Erald Troja and Spiridon Bakiras. Privacy-preserving location-aware mobile advertisement. *In-submission* [80].

REFERENCES

[1] Gergely Acs and Claude Castelluccia. I have a dream! (differentially private smart metering). In *Information Hiding*, pages 118–132, 2011.

[2] Ian F Akyildiz, Won-Yeol Lee, and Kaushik R Chowdhury. CRAHNs: Cognitive radio ad hoc networks. *Ad Hoc Networks*, 7(5):810–836, 2009.

[3] Matt Blaze, John Ioannidis, Angelos D Keromytis, Tal G Malkin, and Avi Rubin. Anonymity in wireless broadcast networks. *International Journal of Network Security*, 8(1):37–51, 2009.

[4] Dan Boneh. The decision Diffie-Hellman problem. In *Algorithmic Number Theory*, pages 48–63. 1998.

[5] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In *TCC*, pages 325–341. 2005.

[6] Felix Brandt. Efficient cryptographic protocol design based on distributed el gamal encryption. In *Information Security and Cryptology (ICISC)*, pages 32–47. 2006.

[7] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In *EUROCRYPT*, pages 402–414, 1999.

[8] Claude Castelluccia, Aldar CF Chan, Einar Mykletun, and Gene Tsudik. Efficient and provably secure aggregation of encrypted data in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 5(3):20, 2009.

[9] Matteo Cesana, Francesca Cuomo, and Eylem Ekici. Routing in cognitive radio networks: Challenges and solutions. *Ad Hoc Networks*, 9:228–248, May 2011.

[10] Supriyo Chakraborty, Kasturi Rangan Raghavan, Matthew P Johnson, and Mani B Srivastava. A framework for context-aware privacy of sensor data on mobile systems. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, page 11. ACM, 2013.

[11] T-H Hubert Chan, Elaine Shi, and Dawn Song. Privacy-preserving stream aggregation with fault tolerance. In *Financial Cryptography and Data Security*, pages 200–214. 2012.

[12] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[13] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

[14] Delphine Christin. Privacy in mobile participatory sensing: Current trends and future challenges. *Journal of Systems and Software*, 2015.

[15] CNN. *Google fires engineer for privacy breach*, 2010. `http://www.cnn.com/2010/TECH/web/09/15/google.privacy.firing/index.html?hpt=T2`.

[16] Claudia Cormio and Kaushik R Chowdhury. A survey on MAC protocols for cognitive radio networks. *Ad Hoc Networks*, 7(7):1315–1329, 2009.

[17] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology*, pages 10–18, 1985.

[18] Zekeriya Erkin and Gene Tsudik. Private computation of spatial and temporal power consumption with smart meters. In *ACNS*, pages 561–577, 2012.

[19] Kenneth Ezirim, Shamik Sengupta, and Erald Troja. (multiple) channel acquisition and contention handling mechanisms for dynamic spectrum access in a distributed system of cognitive radio networks. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 252–256. IEEE, 2013.

[20] Kenneth Ezirim, Erald Troja, and Shamik Sengupta. Sustenance against rl-based sybil attacks in cognitive radio networks using dynamic reputation system. In *Military Communications Conference, MILCOM 2013-2013 IEEE*, pages 1789–1794. IEEE, 2013.

[21] FCC. Third memorandum opinion and order. pages 12–36, 2012. URL `http://transition.fcc.gov/Daily_Releases/Daily_Business/2012/db0405/FCC-12-36A1.pdf`.

[22] FCC. Television band devices. page 11, 2013. URL `http://www.ecfr.gov/cgi-bin/text-idx?c=ecfr&rgn=div6&view=text&node=47:1.0.1.1.16.8&idno=47#47:1.0.1.1.16.8.237.7`.

[23] FCC. White space database administrators guide. pages 3–10, 2013. URL `http://www.fcc.gov/encyclopedia/white-space-database-administrators-guide`.

[24] FCC. Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies. *FCC Report and Order*, March 20, 2005.

[25] Julien Freudiger, Mohammad Hossein Manshaei, Jean-Pierre Hubaux, and David C Parkes. On non-cooperative location privacy: A game-theoretic analysis. In *ACM CCS*, pages 324–337, 2009.

[26] Sheng Gao, Jianfeng Ma, Weisong Shi, Guoxing Zhan, and Cong Sun. Trpf: A trajectory privacy-preserving framework for participatory sensing. *Information Forensics and Security, IEEE Transactions on*, 8(6):874–887, 2013.

[27] Zhaoyu Gao, Haojin Zhu, Yao Liu, Muyuan Li, and Zhenfu Cao. Location privacy in database-driven cognitive radio networks: Attacks and countermeasures. In *IEEE INFOCOM*, pages 2751–2759, 2013.

[28] Gartner. *Gartner Says Worldwide PC, Tablet and Mobile Phone Shipments to Grow 5.9 Percent*, 2013. `http://www.gartner.com/newsroom/id/2525515`.

[29] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *ACM STOC*, volume 9, pages 169–178, 2009.

[30] Craig Gentry and Zulfikar Ramzan. Single-database private information retrieval with constant communication rate. In *ICALP*, pages 803–815. 2005.

[31] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. Private queries in location based services: Anonymizers are not necessary. In *ACM SIGMOD*, pages 121–132, 2008.

[32] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *ACM STOC*, pages 218–229, 1987.

[33] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *ACM STOC*, pages 365–377, 1982.

[34] Jens Groth. Efficient maximal privacy in boardroom voting and anonymous broadcast. In *Financial Cryptography*, pages 90–104, 2004.

[35] Jens Groth, Aggelos Kiayias, and Helger Lipmaa. Multi-query computationally-private information retrieval with constant communication rate. In *PKC*, pages 107–123. 2010.

[36] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *ACM MobiSys*, pages 31–42, 2003.

[37] Saikat Guha, Bin Cheng, and Paul Francis. Privad: Practical privacy in online advertising. In *NSDI*, 2011.

[38] Hamed Haddadi, Pan Hui, and Ian Brown. MobiAd: private and scalable mobile advertising. In *ACM MobiArch*, pages 33–38, 2010.

[39] Feng Hao and Piotr Zieliński. A 2-round anonymous veto protocol. In *Security Protocols*, pages 202–211, 2009.

[40] Michaela Hardt and Suman Nath. Privacy-aware personalization for mobile advertising. In *ACM CCS*, pages 662–673, 2012.

[41] Simon Haykin, D.J. Thomson, and J.H. Reed. Spectrum sensing for cognitive radio. *Proceedings of the IEEE*, 97(5):849–877, 2009.

[42] Wenbo He, Xue Liu, Hoang Nguyen, Klara Nahrstedt, and TT Abdelzaher. PDA: Privacy-preserving data aggregation in wireless sensor networks. In *IEEE INFOCOM*, pages 2045–2053, 2007.

[43] Leping Huang, Kanta Matsuura, Hiroshi Yamane, and Kaoru Sezaki. Enhancing wireless location privacy using silent period. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 2, pages 1187–1192. IEEE, 2005.

[44] Leping Huang, Hiroshi Yamane, Kanta Matsuura, and Kaoru Sezaki. Silent cascade: Enhancing location privacy without communication qos degradation. In *Security in pervasive computing*, pages 165–180. Springer, 2006.

[45] Min-Shiang Hwang, Chia-Hui Wei, and Cheng-Yee Lee. Privacy and security requirements for rfid applications. *Journal of Computers*, 20:55–60.

[46] IETF. Tcp congestion control. 1999. URL `https://www.ietf.org/rfc/rfc2581.txt`.

[47] Marek Jawurek and Florian Kerschbaum. Fault-tolerant privacy-preserving statistics. In *PETS*, pages 221–238, 2012.

[48] Ari Juels. Targeted advertising... and privacy too. In *CT-RSA*, pages 408–424. 2001.

[49] Taeho Jung, Xiang-Yang Li, and Meng Wan. Collusion-tolerable privacy-preserving sum and product calculation without secure channel. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 12(1):45–57, 2015.

[50] Ibrahim Kamel and Christos Faloutsos. On packing R-trees. In *ACM CIKM*, pages 490–499, 1993.

[51] Aggelos Kiayias and Moti Yung. Non-interactive zero-sharing with applications to private distributed decision making. In *Financial Cryptography*, pages 303–320, 2003.

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

[53] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *IEEE FOCS*, pages 364–373, 1997.

[54] Byoungyoung Lee, Jinoh Oh, Hwanjo Yu, and Jong Kim. Protecting location privacy using location semantics. In *ACM SIGKDD*, pages 1289–1297, 2011.

[55] Chun-Ta Li and Min-Shiang Hwang. A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks. *Information Sciences*, 181(23): 5333–5347, 2011.

[56] Qinghua Li, Guohong Cao, and Thomas F. La Porta. Efficient and privacy-aware data aggregation in mobile sensing. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 11(2):115–129, 2014.

[57] Shuai Li, Haojin Zhu, Zhaoyu Gao, Xinping Guan, Kai Xing, and Xuemin Shen. Location privacy preservation in collaborative spectrum sensing. In *IEEE INFOCOM*, pages 729–737, 2012.

[58] Y. Lindell and B. Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1):59–98, 2009.

[59] Helger Lipmaa. An oblivious transfer protocol with log-squared communication. In *Information Security*, pages 314–328. 2005.

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

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

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

[63] Joseph Mitola III. Cognitive radio: An integrated agent architecture for software defined radio. *Doctoral Dissertation, KTH, Stockholm, Sweden*, May 2000.

[64] Ana Nika, Zengbin Zhang, Xia Zhou, Ben Y Zhao, and Haitao Zheng. Towards commoditized real-time spectrum monitoring. In *Proceedings of the 1st ACM Workshop on Hot Topics in Wireless*, pages 25–30. ACM, 2014.

[65] NTIA. United states frequency allocations – the radio spectrum. 2003. URL `http://www.ntia.doc.gov/files/ntia/publications/2003-allochrt.pdf`.

[66] Rafail Ostrovsky and William E Skeith III. Private searching on streaming data. In *CRYPTO*, pages 223–240, 2005.

[67] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238, 1999.

[68] Stavros Papadopoulos, Spiridon Bakiras, and Dimitris Papadias. pCloud: A distributed system for practical PIR. *IEEE Transactions on Dependable and Secure Computing*, 9 (1):115–127, 2012.

[69] Stavros Papadopoulos, Aggelos Kiayias, and Dimitris Papadias. Exact in-network aggregation with integrity and confidentiality. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 24(10):1760–1773, 2012.

[70] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO*, pages 129–140, 1991.

[71] Torben P. Pedersen. A threshold cryptosystem without a trusted party. In *EUROCRYPT*, pages 522–526, 1991.

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

[73] Fudong Qiu, Fan Wu, and Guihai Chen. Slicer: A slicing-based k-anonymous privacy preserving scheme for participatory sensing. In *Mobile Ad-Hoc and Sensor Systems (MASS), 2013 IEEE 10th International Conference on*, pages 113–121. IEEE, 2013.

[74] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[75] Elaine Shi, T-H Hubert Chan, Eleanor G Rieffel, Richard Chow, and Dawn Song. Privacy-preserving aggregation of time-series data. In *NDSS*, 2011.

[76] Jing Shi, Rui Zhang, Yunzhong Liu, and Yanchao Zhang. Prisense: privacy-preserving data aggregation in people-centric urban sensing systems. In *IEEE INFOCOM*, pages 1–9, 2010.

[77] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[78] Wee Lum Tan, Fung Lam, and Wing Cheong Lau. An empirical study on the capacity and performance of 3G networks. *IEEE Transactions on Mobile Computing*, 7(6):737–750, 2008.

[79] Vincent Toubiana, Arvind Narayanan, Dan Boneh, Helen Nissenbaum, and Solon Barocas. Adnostic: Privacy preserving targeted advertising. In *NDSS*, 2010.

[80] Erald Troja and Spiridon Bakiras. Privacy-preserving location-aware mobile advertisement. *In-submission.*

[81] Erald Troja and Spiridon Bakiras. Leveraging P2P interactions for efficient location privacy in database-driven dynamic spectrum access. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 453–456. ACM, 2014.

[82] Erald Troja and Spiridon Bakiras. Efficient location privacy for moving clients in database-driven dynamic spectrum access. In *Proceedings of the 24th International Conference on Computer Communications and Networks*. IEEE, 2015.

[83] Erald Troja and Spiridon Bakiras. Leveraging P2P interactions for efficient location privacy in database-driven dynamic spectrum access. *International Journal of Network Security*, 17(5):569–579, 2015.

[84] Erald Troja, Kenneth Ezirim, and Suman Bhunia. Route aware dynamic channel scheduling and selection for multi-hop cognitive radio networks. In *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, pages 1–5. IEEE, 2013.

[85] Erald Troja, Kenneth Ezirim, Shamik Sengupta, and Michael Hannon. Performance evaluation of RODEO: Route degradation optimization for the multi-hop dynamic spectrum access networks. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 232–236. IEEE, 2013.

[86] Jonathan Trostle and Andy Parrish. Efficient computationally private information retrieval from anonymity or trapdoor groups. In *Information Security*, pages 114–128. 2011.

[87] Janice Y Tsai, Patrick Gage Kelley, Lorrie Faith Cranor, and Norman Sadeh. Location-sharing technologies: Privacy risks and controls. *ISJLP*, 6:119, 2010.

[88] Khuong Vu, Rong Zheng, and Jie Gao. Efficient algorithms for k-anonymous location privacy in participatory sensing. In *IEEE INFOCOM*, pages 2399–2407, 2012.

[89] WSJ. *Mobile-Ad Spending Leaps, but Trails User Growth*, 2014. `http://www.wsj.com/articles/mobile-ad-spending-leaps-but-trails-user-growth-1405969018`.

[90] Zhiqiang Yang, Sheng Zhong, and Rebecca N Wright. Privacy-preserving classification of customer data without loss of accuracy. In *SDM*, pages 92–102, 2005.

[91] Zongkai Yang, Geng Cheng, Wei Liu, Wei Yuan, and Wenqing Cheng. Local coordination based routing and spectrum assignment in multi-hop cognitive radio networks. *Mobile Networks and Applications*, 13:67–81, April 2008.

[92] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *IEEE FOCS*, pages 162–167, 1986.

[93] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: driving directions based on taxi trajectories. In *ACM GIS*, pages 99–108, 2010.

[94] Tan Zhang, Ashish Patro, Ning Leng, and Suman Banerjee. A wireless spectrum analyzer in your pocket. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 69–74. ACM, 2015.

[95] Wei Zhang, R.K. Mallik, and K. Letaief. Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks. *IEEE Transactions on Wireless Communications*, 8(12):5761–5766, 2009.

[96] Qianchuan Zhao, Stefan Geirhofer, Lang Tong, and Brian M Sadler. Optimal dynamic spectrum access via periodic channel sensing. In *IEEE WCNC*, pages 33–37, 2007.

[97] Yu Zheng, Longhao Wang, Ruochi Zhang, Xing Xie, and Wei-Ying Ma. GeoLife: Managing and understanding your past life over maps. In *IEEE MDM*, pages 211–212, 2008.

[98] Zhou Zhi and Yow Kin Choong. Anonymizing geographic ad hoc routing for preserving location privacy. In *Distributed Computing Systems Workshops, 2005. 25th IEEE International Conference on*, pages 646–651. IEEE, 2005.