ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 4, No. 1, 27-39 (2019)

www.astesj.com

Special Issue on Recent Advances in Engineering Systems

ASTESJ

ISSN: 2415-6698

A Practical PIR-based Scheme for Discovering Nearby Places for Smartphone Applications

Maryam Hezaveh*, Carlisle Adams

School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada

ARTICLE INFO

Article history:
Received: 10 December, 2018
Accepted: 14 January, 2019
Online : 20 January, 2019

Keywords:
Private Information Retrieval
Privacy
Smartphone applications
Location-Based Services

ABSTRACT

We present a privacy-preserving approach for discovering nearby places of interest to Alice. In this approach, the proposed protocol allows Alice to learn whether there is any place that she is looking for near her. However, the location-based service (LBS) that tries to help Alice to find nearby places does not learn Alice's location. Alice can send a request to the LBS database to retrieve nearby places of interest (POIs) without the database becoming aware of what Alice fetched by using private information retrieval (PIR). The common criticism of previous PIR approaches is that they are not practical for smartphones with limited processing power, memory, and wireless bandwidth due to the computational overhead. Therefore, the main focus of this work is to propose a scheme to reduce the computation cost on the client-side to make PIR appropriate and practical for the smartphone environments, and then apply the proposed PIR to LBS applications. We have implemented our protocol in Percy++ to evaluate its performance over a commercial-grade database of POIs. Our implementation results demonstrate that our approach has faster decode and retrieve time for the nearby POIs on smartphones compared with current similar work.

1. Introduction

Location-based services (LBS) are information services that offer various types of applications based on the location of the user, such as identifying a location of a person or object or place, weather service, parcel and vehicle tracking, etc. LBS retrieves the location of the user from the user's mobile phone via global positioning system (GPS), cell tower triangulation, or wireless local area network (WLAN). LBSs might be helpful to mobile users for safety services. For example, it is beneficial if emergency services can find information about the location details of a user who is in danger. However, users may not be aware that their location information might be shared with other third parties and could be misused, and it could be diverted as a tracking tool. Therefore, the main goal for research communities in this field is to protect the user's location while they are using LBSs.

During the last two decades, privacy-preserving protocols for location-based services have been introduced based on non-cryptographic and cryptographic approaches. Non-cryptographic approaches use trusted third parties to maintain the user's privacy, such as "dummy locations," "K-anonymity" and "cloaking" approaches. Reviews of these approaches and their drawbacks can be found in [1, 2, 3, 4, 5]. The main disadvantage of these approaches is that the user has to trust and send her information to the trusted third party, which as we mentioned is unacceptable in LBS applications.

Our main goal in this paper is to help LBS users search on their smartphone for particular places of interest (POIs) while keeping their location private from the location-based service provider (LBSP). For example, the user sends her request to the LBS application to find a nearby restaurant, gas station, or ATM. This location could be the exact location of the user or a location to which she wants to travel in the future. These types of applications are especially useful for people who travel or have moved to a new city. A wide range of LBS nearby places applications have been released recently, such as Facebook Nearby Places, AroundMe, NearBy Places, Yelp, FourSqure and Places to help users to identify their nearby locations quickly.

Modern multi-core mobile devices have high-performing processors that are appropriate for cryptographic tasks which can enable location privacy to the LBS applications. Unfortunately, the processing units frequently consume a significant amount of energy, which causes a reduction in the battery life of smartphones. In addition, smartphones devices often have limited bandwidth and memory [6]. Therefore, downloading an entire database for finding POIs around Canada and the U.S., which easily can include more

*Maryam Hezaveh, University of Ottawa, mheza028@uottawa.ca

https://dx.doi.org/10.25046/aj040104

than ten million entries with respect to a typical commercial POI database (see Appendix A) [7] and require 3 to 4 GB of data storage, is obviously not practical. Furthermore, updating results periodically to make sure they are accurate enough, bandwidth limitation and data usage limitation of smartphones are other important factors which we should keep in mind when we are offering a cryptographic solution for LBS applications with respect to the privacy of the mobile user's location.

1.1. Motivation and Threat Model

Enormous enthusiasm for geographical referencing of individual information is apparent on the web these days. The majority of people currently use smartphones with many complex sensors closely connected to their daily activities. Most of these smartphones have a high-precision localization sensor such as a GPS receiver. GPS devices allow people to tag photographs and occasions and to track their mobility. Moreover, the number of sensors in our environment which interact with smartphones has increased. Although most people like the convenience of using these personal communication devices, there is an inherent trade-off between convenience and privacy. Clients might not be completely aware of exactly how the data about their location is utilized and by whom and what information about their location is being gathered, and subsequently, clients can disregard the potential risk that can happen by using their location information.

Location-aware capabilities allow the service providers to offer different types of application to their users such as the ability to share their location with their friends and other users and to geo-reference their posts. In this way, clients can utilize the location identifier to search and browse for different resources. An essential key for providing these services is to gather real-time location information on clients and additionally, other logical data including client relationships, activities and client provided content perhaps during long intervals of time. Specifically, in nearby location applications, service providers are not only able to collect clients' location information, but are also able to gather personal information by offering clients to write their experience and opinions with regard to reviews and tips on the visited place. Subsequently, clients' historical location data can be identified with relevant and semantic data freely accessible online and can be utilized to discover individual and sensitive data about clients and to develop comprehensive client profiles. The user activities, relationships, interests and mobility patterns could be extracted from these profiles. Although these location-based profiles may be considered helpful to improve and personalize the quality of applications for the clients, they can potentially be utilized for unwanted purposes and can cause different levels of privacy threats. Users' mobility tracks are not only a collection of locations on a map. The content of these tracks includes the users' interests, activities, habits, and relationships. It may also disclose users' private information and secrets. It can expose the users to undesirable commercial and spams, or even threat of physical harm. All of these imply that the negative side-effects of lacking location privacy have increased.

The main aim of this paper is to protect the users' location privacy against a passive adversary, active adversary and malicious service providers while they are using LBSs. We consider the following threats in our architecture:

Passive adversary. A malicious external observer or a malicious LBSP who has access to the data that passes between the user and the database on the communication channel but cannot change the data.

Active adversary. A malicious external observer who has access to data that passes between the user and the database on the communication channel and can insert, modify or delete data.

Malicious Service provider. A malicious server refers to an LBSP that tries to modify, delete or insert new messages in response to the user.

Users should have the privilege of controlling the amount of information (about their location) that is revealed and shared with others. This can be achieved in different ways such as users have a right to choose not to share their location information to untrusted applications, legislating privacy policies to force organizations and service providers to protect their users' location privacy, and designing a system in a privacy-preserving manner so it does not disclose users' location information to others.

1.2. Our Contributions and Assumptions

This current paper is an extension of a paper originally presented in [8]. We first explain our proposed block-based PIR scheme for smartphone applications [8] and then as an extension we apply our proposed PIR scheme to the LBS application. Our privacy-preserving protocol for LBS helps Alice to search for specific nearby POIs on her smartphone by sending a query to the location based service provider (LBSP) over a wireless network. In this scenario, the proposed protocol allows Alice to learn whether there is any place that she is looking for near her. However, the location-based service (LBS) that tries to help Alice to find nearby places does not learn Alice's location. Alice can send a request to the LBS database to retrieve nearby places of interest (POIs) on her smartphone without the database becoming aware of what Alice fetched by using our practical PIR scheme. The LBS server retrieves the query from the database, and returns the results to Alice containing the specific requested POIs type found in the requested location. In order to achieve this, our protocol must fulfill all of the following requirements, as also required in [9]:

1. The LBS server must not learn the exact location of the user. It might only identify a area that is large enough to satisfy the user's privacy in terms of area and the number of POIs it contains.
2. The proposed protocol must have no third parties between the user and the server.
3. The implementation must be computationally practical for resource-constrained hardware such as a smartphone.

4. The proposed approach cannot depend on trusted hardware that does not generally exist on a commercial smartphone.

Our cryptographic approach is based on private information retrieval (PIR) for secure LBS applications that identify nearby places. PIR allows the user to fetch her required information from the database without leaking which information is fetched [10]. The POI database is labeled by the location of POIs; therefore, the LBS server is able to retrieve the POIs depending on the user's location of interest in the requested query. PIR solves most of the previous problems associated with non-cryptographic approaches in LBS. PIR approaches do not have the privacy vulnerabilities of k-anonymity or cloaking, such as single point of attack of their anonymizer or server which tries to help them to apply k-anonymity or generate an obfuscation area. As a result, the information of the user location remains private and secure from all kind of the passive adversary, active adversary and service provider by using PIR approaches.

During the past two decades, various types of PIR-based approaches have been introduced. The common criticism of previous PIR approaches is that they are not practical for smartphones with limited processing power, memory, and wireless bandwidth due to the computational overhead [11, 12]. We ensure that the proposed cryptographic PIR approach is practical for smartphone applications. Based on [13], there are five main time elements that influence the speed of the PIR query:

1. the amount of time that it takes for the client to create a query which has to be private.
2. the amount of communication time that it takes to send the query to the server(s).
3. the amount of time for the server(s) to apply the query to the database.
4. the amount of communication time that it takes to send the response from the server(s) to the client.
5. the amount of time that it takes for the client to decode the response(s) and retrieve the results.

Our approach expands [9] idea of applying a cloaking area to reduce these five factors. Moreover, our approach reduces the amount of time required for the client to process the response and retrieve the results of decoding on the smartphone by approximately 50% compared to [9] by applying the POI types idea to block-based PIR. Reducing the decode time is valuable in our application to satisfy the fifth requirement, so that it can be used on modern smartphones' hardware. The processing cost on the server side is similar to [9], to preserve the privacy of the user's location. Our proposed protocol can be made to support all types of block-based PIR schemes.

In our proposed approach, the identity of the user is not hidden from the service provider, as the results have to be returned to the user. However, if the user wants to keep her identity hidden from LBS, she can use an onion routing technique, such as Tor [14]. Note that keeping the user's location private has priority in an LBS application over keeping the user's identity hidden from LBS, because if LBS knows the user's location, it is quite easy to identify the user. We should mention that a mobile communications operator is constantly aware of the location of the user based on the cell tower. Therefore, we assume that this operator does not collude with the LBSP.

1.3. Organization of This Work

The rest of this paper is structured as follows. Section 2 presents an overview of previous work regarding PIR schemes and LBS schemes. Section 3 describes the details of our PIR scheme. Section 4 explains the details of our privacy-preserving protocol for LBS. The threat model and the security analysis of our proposed protocol are discussed in Section 5. Section 6 gives an overview of our implementation and compares it with previous work. The limitations of our proposed protocol are discussed in Section 7, and finally Section 8 concludes our paper.

2. Related Work

For greater understanding, we first review the definition of PIR and give a brief overview of different types of PIR schemes. Then, we provide a review of PIR-based approaches for the users' location privacy in LBS applications.

2.1 Review of Private Information Retrieval (PIR)

These days, users are increasingly aware of the privacy requirements of their data in their online activities. But is it actually possible to keep the user's query contents private while she issues a request to online applications? The first answer that comes to your mind when you think about this problem is that the user can send her request to the online application via Tor and communicate over the Tor network [14]. Here, the server has no clue who sent the request for the data; however, in order to fetch the requested data from the database, the server has to be able to access the content of query. Therefore, Tor is not a good option to solve our problem. The main problem that we need to solve is to let a user to send her query to the database without sharing what she searched for. In this scenario, we are trying to protect the content of the query, rather than the identity of the user. Private information retrieval (PIR) is a cryptographic technique that solves the matter of permitting the user to query a database while the content of the user's query is hidden from the database. The need for PIR schemes has been demonstrated in real online activities, such as location-based services, social networks, online research, etc. [9].

In 1995, [10] first introduced the problem of Private Information Retrieval (PIR). Looking at the trivial solution [10] of transferring the entire database to the user to be locally queried, highlights interesting properties. First, it delivers perfect privacy. Second, no information about query or response is leaked, since neither of these are sent across the wire. On the other hand, this approach yields high communication overhead: the size of the whole database. Goldberg [15] presented three important requirements for PIR: privacy, non-triviality, and correctness. For privacy, the database should learn neither the query input nor the

database block retrieved. For non-triviality, communication cost between the client and the server should be less the trivial limit of $O(n)$, where n is the number of bits in the database as seen above. For correctness, the received data from the database must satisfy the user's query. Another requirement which is not considered in most of the previous work for PIR is implementation efficiency. Most of the previous work tried to reduce the communication overhead rather than the computational overhead [16, 17]. This inattention to the computational complexity has caused the introduction of PIR schemes that are not practical for resource-constrained hardware, such as smartphones.

In [10], the author defined the first non-trivial PIR scheme. In 2004, Gasarch [18] described it simply as follows.

Definition 2.1. A one-round k-databases Private Information Retrieval (PIR) scheme with x$\in \{0,1\}^n$ is defined as follows [10, 18].

1. A user wants to find x_i. There exists k databases which all have the same copy of $x = x_1 \dots x_n$. The DBs do not collude with each other.
2. The user flips coins and the combination of the coin flips and i, produces query strings $q_1 \dots q_k$. She sends the query, q_j, to database DB_j.
3. For all j queries, where $1 \leq j \leq k$, DB_j returns an answer string $ANS_j(q_j)$.
4. The user computes x_i using the value of the $ANS_j(q_j)$, the coin flips, and i.

The cost of the defined PIR scheme is $\sum_{j=1}^{k} |q_j| + |ANS_j(q_j)|$.

Computational PIR (CPIR): The first type of PIR protocols assumes that the adversary and the server(s) have access to limited computational capability to guaranty the user's privacy. Therefore, to breach the security of these protocols, the adversary has to solve a problem which is hard to solve with its limited computational capability. This kind of assumption is usual for cryptography, security, and privacy schemes.

In 1995, [10] proved that it is impossible to have a single-database PIR in the information theoretic security sense. In 1997, [19, 20] proposed the first CPIR to prove that the communication complexity of PIR can be reduced if we want to achieve computational privacy, and we are not willing to achieve information theoretic privacy. In the same year, Kushilevitz and Ostrovsky [21] presented a CPIR protocol which has the same assumption for the computational capability for the adversary, but it uses a single server. Their protocol was the first single-server CPIR. It is based on the Quadratic Residuosity problem that is considered to be difficult to solve. The main advantage of single-server CPIR protocols is that by using the CPIR recursively, the communication complexity of PIR can be improved. Later, different types of single-server CPIR were proposed which tried to reduce the communication cost of PIR, for example, φ-hiding problem [22, 23], the presence of one-way trapdoor permutations [24], Pailler homomorphic encryption [25], and the Hidden Lattice problem [16].

In [17], the author proved that none of the previous CPIR schemes were practical, given certain realistic assumptions at the time. However, in 2016, [26] introduced XPIR. They showed that by using lattice-based cryptography, CPIR is of practical value and the conclusion of [17] is no longer valid.

Information Theoretic PIR (IT-PIR): In Information theoretic privacy even if an adversary has unlimited computational capability, he cannot compromise the privacy of the user. In 1995, [10] showed that any single-server IT-PIR scheme must have communication cost at least that of the trivial protocol. Therefore, IT-PIR protocols assume that if you have $k \geq 2$ non-cooperating servers, and each of these servers has a copy of the database, then there exist PIR schemes which achieve complete information theoretic security. Following [10], different types of IT-PIR were proposed which tried to improve [10], such as [9, 15, 27, 28].

By using the idea of multiple servers, we improved the robustness of the PIR, but this can affect privacy if there exists non-responsive servers or/and malicious servers [15, 27]. To handle this issue, Goldberg [15] proposed the privacy threshold in which the total number of the servers must be greater than the privacy threshold. As a result, in order to set a privacy threshold, we need to provide extra responding servers.

Trusted Hardware PIR: The trusted hardware-based PIR is first introduced by [29] in 2006. The trusted hardware-based PIR uses the idea of a tamper-resistant CPU, which is connected to the server and is trusted by the user. The user sends her query to this CPU, where her query is hidden from the server. In this scenario, the CPU is the one who is responsible to fetch the requested information from the database and sends back the results to the user. These types of PIR achieve the low computation and communication costs, but the trusted hardware PIR architecture is secure only if the user can trust the hardware.

Hybrid PIR: In [13], the researcher proposed a hybrid PIR that was a combination of CPIR and IT-PIR to reduce communication costs. Their goal was to combine the positive features of CPIR and IT-PIR to reduce the negative features of each. To achieve a lower bound for both computation and communication costs, they merged the recursion property of CPIR (single-server) approaches and the low computation and communication complexity property of IT-PIR (multiple-server) approaches.

2.2 Review of the PIR-based scheme for Nearby Places

One of the motivations for developing useful and practical PIR schemes is to protect the users' private information while they are using mobile devices with positioning capabilities. In a stationary desktop scenario, when a user tries to query the database or the remote server, the primary concern is leaking information about the query's content. However, in an LBS scenario, when a user queries the LBS server, her location is also revealed to the LBS server. Here, the problem with location privacy is preserving the privacy of the user's real location when she is using the LBS while providing the most precise and acceptable response.

Many of the previous problems of privacy preserving protocols for LBS that we encountered were solved by introducing PIR-based LBS protocols. The idea is to let the user send a query to the LBS server without disclosing her actual location by the PIR scheme. This query typically consists of POIs, which includes a description of the POI and its geographic location.

Most of the existing works which tried to apply PIR to location-based services were based on secure hardware, with a secure coprocessor at the LBS server [3, 5, 30, 31]. The idea of using the secure hardware-based PIR in LBS was first proposed by Hengartner [3]. This hardware performs the trusted computing to hide the user's location from the LBSP. Recent work regarding secure hardware PIR was proposed by [30]. Their PIR technique was similar to [31], however it offered better efficiency, and it was more practical for large datasets. All proposed solutions for secure-hardware PIR claim that the trusted hardware-based PIR method is the only practical PIR scheme [30, 31]. The main disadvantage of all secure hardware PIR schemes is that the proposed architectures are secure only if the user can trust the hardware.

The common criticism of other PIR approaches for location privacy is that the computational overhead is not acceptable and practical for resource-constrained hardware such as a smartphone [11, 12]. In 2008, In [1], the author proposed the first PIR-based approach for location privacy, without using a third party. Their proposed protocol used the idea of the trade-off between efficiency and privacy as defined in [32]. In [1], the researcher proposed a single PIR request for each query approach. In their approach, all queries were indistinguishable, and it was able to achieve strong location privacy. Their proposed protocol included two steps to protect the user's query and information about her location. In the first step, the server and the user engaged in a protocol, which is based on Paillier encryption [33], to determine the index of the user's location cell, without releasing the location to the server. The user uses PIR to retrieve the query results for the target cell in the second step. The advantages of the Ghinita protocol are the nondisclosure of location information and its security for both mobile and stationary users against correlation attacks.

In [5], the author described three drawbacks to [1] protocol. First, it focuses on the nearest neighbor queries. Second, it scans the entire database linearly for each query. Third, it has a high communication complexity. Additionally, the protocol is secure if the privacy of the user is a concern and LBS is not able to learn the user's query, but it is not symmetric for LBS's database privacy since the user can infer the data that are in the same column as her query.

Later, in [9], the author proposed a hybrid solution combining PIR and cloaking to protect the user's privacy without using trusted computing. Their idea of using cloaking reduces the computational cost of PIR and makes it more practical. The user's location privacy relies on the size of the cloaking area. Their PIR approach supports all types of PIR schemes (block-based). Our proposed PIR protocol expands on [9] idea. However, we focus on reducing the computational complexity on the client side. We explain our proposed protocol in detail in the next section.

3. The Proposed PIR Scheme

Here we present our block-based PIR for location privacy in mobile phone applications. Our solution uses partial queries [10] to reduce communication and computation complexity. Moreover, we structure the database to optimize client computations. This has benefit in our mobile scenario in which the clients (possibly smartphones or IoT sensors) have constrained computational power. In our approach, the user retrieves the exact category of the data, which saves on data processing on the resulting sets. These savings on result set size in turn impact any decode, decrypt, or homomorphic operations which must occur to obtain a result. As these are cryptographic operations, the benefit in result set size reduction is material. Note that our approach is suitable for all applications that need to protect users' privacy while they are searching for data in a database (it is not restricted to just LBS applications).

3.1. Preliminaries

Our proposed protocol can be made to support all types of block-based PIR schemes. We illustrate its usage using multiple server IT-PIR [15] and Shamir secret sharing [34]. As such, the user's query is split into l shares which are then transferred to k servers. This results in communications and computation benefit which we analyze in Section 6. The protocol is robust to byzantine situations in which servers (either malicious or in a service degradation scenario) may fail to respond or may respond with information containing errors.

Our approach to reduction of client computation cost uses the idea of trading off privacy for better performance [9]. In [9], the level of desired-privacy is adjustable and is proportionally related to the number of data items that the database PIR server must process to respond to the client. We extend and improve on this approach in three ways.

First, we divide the database into classes, with each class categorized based on a sub-type of data to be queried. The server returns exactly the subset of the database which pertains to the queried category. By reducing result set size, the client benefits in a number of ways. It is no longer necessary to filter the response data. In addition, the aggregate cost of cryptographic operations, such as decryption or homomorphic computation, is reduced.

Second, If a sub-type has a higher amount of data, a data traffic cost will be higher because of the result size and it will cause a slower response time. On the other hand, if in another sub-type the amount of data is extremely low, it will minimize the result size and lower data traffic. These could help the server to guess the user's sub-type of interest with a high degree of confidence and lead to a loss of privacy. To tackle this problem, our approach equalizes the amount of data in each row of the sub-type in a specific class by adding "null" to all the sub-types which are not equal to the maximum sub-type size in that class. The main advantage of our "null" solution is that if the user is looking for a sub-type with less data, the PIR computation overhead on the

client-side is reduced (when receiving the first null in the decoding process, the computation process stops), and if the user is looking for a sub-type with more data, the PIR computation cost on the client-side increases, without losing privacy.

Finally, in our approach the amount of data is different in different classes (see Figure 1), unlike [9]. As a result, our PIR computation cost depends on the specific class that the user requests. If the user searches for a class with more data, the PIR computation cost increases. If the user searches for a class with less data, the PIR computation cost is reduced.

By considering these three improvements, if the user sends a query to the database for the sub-type of data in each class, the response which is returned to the user not only needs less computation time for decoding, but also does not need to be filtered on the client side to remove non-requested data.

Class	Category Types					
Class-0 00	Sub-type 0 0000	data1	data2			
	Sub-type 1 0001	data 1	-			
	Sub-type 2 0010	data1	data2			
	Sub-type 3 0011	-	-			
	Sub-type 4 0100	-	-			
	Sub-type 5 0101	data1	data2			
	Sub-type 6 0110	data1	data2			
	Sub-type 7 0111	-	-			
	Sub-type 8 1000	-	-			
	Sub-type 9 1001	data1	-			
Class-1 01	Sub-type 0 0000	-				
	Sub-type 1 0001	-				
	Sub-type 2 0010	-				
	Sub-type 3 0011	-				
	Sub-type 4 0100	-				
	Sub-type 5 0101	-				
	Sub-type 6 0110	-				
	Sub-type 7 0111	-				
	Sub-type 8 1000	-				
	Sub-type 9 1001	-				
Class-2 10	Sub-type 0 0000	-	-	-	-	-
	Sub-type 1 0001	data1	data2	data3	data4	data5
	Sub-type 2 0010	-	-	-	-	-
	Sub-type 3 0011	data1	data2	-	-	-
	Sub-type 4 0100	data1	-	-	-	-
	Sub-type 5 0101	data1	data2	data3	data4	data5
	Sub-type 6 0110	data1	-	-	-	-
	Sub-type 7 0111	data1	data2	data3	data4	-
	Sub-type 8 1000	-	-	-	-	-
	Sub-type 9 1001	-	-	-	-	-
…	…					

Figure 1 Sample of the relationship between data, sub-type and classes, as saved in the database.

3.2. The Proposed PIR Scheme

Our PIR protocol has two phases. The first phase is the pre-processing phase in which the whole protocol becomes ready to use, on the server side and also on the client side. In the future, if the client decides to change the level of her privacy, or any changes occur on the server side, this phase can be repeated. The second phase is the execution phase in which the user sends her request to the server. Her request contains the class of data which she searches concatenated with the sub-type category.

Pre-processing Phase contains the following steps:

1. Given a chosen level of the user's privacy, "Class", "sub-type" and "data" category are applied to the database.
2. The "class" and "sub-types" are defined to have a number based on their specific categories. As shown in Figure 1, for example, Class-1 is considered 01 and Sub-type-3 is considered 0011. Note that in this Figure we just showed "10" different "sub-types" for each "class". This depends on the number of different sub-types of data in the database and also on the level of the user's privacy which is applied in step 1.
3. Each database index will be the "class" concatenated with the "sub-type". For example, in Figure 1, the database index for the Class-1||sub-type-3 is considered as 010011.

Execution Phase contains the following steps:

1. The user chooses the sub-type of her interest from the list suggested by the application based on her privacy level. For example, she is looking for Class-1||sub-type-3.
2. The proposed application provides the user's request, which is an index of the database, and sends it to the server in a way that is hidden from the server. In this example, the request is 010011 which refers to Class-1||sub-type-3.
3. The specific row of database is retrieved from the database and the data present in this row are transmitted back to the user.
4. The user decodes the results and the results are shown on her smartphone.

4. The Proposed Privacy-Preserving Protocol for LBS

The main goal of cryptographic protocols in nearby places is to be able to detect nearby places automatically while the user's location privacy is considered in the location-based service (LBS) application. Our proposed protocol uses private information retrieval (PIR) to achieve this purpose.

4.1. Problem Statement

Alice has her location as her secret. Alice wants to use a LBS application to search and find nearby places of interest. We propose a protocol that allows Alice to find nearby places for which she is looking. However, the LBS that helps Alice to find her nearby place does not learn Alice's location. Alice can send a request to the LBS's database to fetch her nearby places of interest without the LBS being aware of what Alice fetched by using private information retrieval (PIR). Most of the previous PIR schemes are not acceptable in LBS applications because of their use of secure hardware. The focus of this section is to solve the PIR-based LBS issues by offering a practical PIR without using secure hardware or a trusted third party and lower the computational cost on the client side in the smartphone's application. At the end of this protocol, the proposed application should list the POIs that meet Alice's search criteria or show her that there is no POI in the selected area.

4.2. The Proposed Privacy-Preserving Protocol for LBS

We first informally describe our proposed protocol via an example. Suppose Alice is located in Ottawa and she wants to look for a specific type of POI, for example a restaurant, near Bank Street. Since she is privacy conscious, she sets her cloaking area to be a 10 km MGRS grid square (see section 4.3). The client

application sends the requested cloaking area to the server. At the same time, the PIR allows the client application to identify which part of the cloaking area has the restaurant, without the server being informed which part is retrieved. All entries in the POI database are indexed by their MGRS block concatenated with the POI type. The row that contains the restaurant(s) is retrieved from the selected MGRS grid square on the database and the results are sent back to Alice. The client application decodes the results and sorts the results, and the nearest restaurants are shown on her phone's local map.

Our protocol follows [9] hybrid solution that uses PIR to preserve the privacy of the user's query and a cloaking scheme in order to make the PIR scheme practical and reduce the computational cost of PIR. The benefits of the hybrid solution are as follows: the location of the user remains secret from the LBSP to a reasonable privacy level chosen by the user without depending on the other users in the selected area; to calculate the cloaking area or cryptographic algorithms we do not need to have a trusted third party; and the computational overhead of the PIR scheme is practical.

Our proposed protocol has two improvements compared to [9]. First, due to the user's request for a specific POI, our proposal categorizes the cloaking area in the database into the POI types. Thus, when the user asks for her POI in her selected cloaking area, the results that are returned to her are of the type that she is looking for. Therefore, our protocol on the client is not required to filter the block of different types of POIs to identify the POI that the user requested. This reduces the computational costs, and saves the battery and data usage on the smartphone. For example, if there are no restaurants near the user, she does not need to wait to decode all POIs in that cloaking area and then filter the restaurant to find that actually the answer is "null".

Second, we propose a new technique based on a static grid-based approach for defining our cloaking area and mapping our POIs to a cloaking area, unlike the approach proposed in [9] which uses the Various-size-grid Hilbert Curve (VHC) technique [35] (see section 4.4).

Our proposed protocol describes how the POI database is initialized and how the protocol generates a cloaking area around the user's exact location, and executes a PIR query on the contents of the requested cloaking area. We name our phases similarly to [9] to highlight the similarities and differences between our phases. Note that each POI consists of 300 bytes that includes longitude and latitude coordinates, name, exact address, the phone number, website address, etc.

The pre-processing phase contains the following steps:

1. An appropriate static grid system is applied on the geolocation plane.
2. POIs are categorized based on their type and saved in the LBS's database.
3. A row of database refers to a cloaking area concatenated with the POI type.

The execution phase contains the following steps:

1. The user selects the area of her interest; it could be her current location as determined through GPS, or some other location that the user may be traveling to in the future.
2. The user selects a preferred level of privacy.
3. The user's corresponding cloaking area is calculated based on the level of her chosen privacy.
4. The user chooses the POI type(s) from the suggested list provided by the client application.
5. The client application sends the cloaking area to the server. Also, the client application identifies which portion of the cloaking area contains the POI type(s), in a way that is hidden from the server.
6. The server receives the request, and finds the database portion corresponding to the cloaking area. A block of rows is retrieved from this portion based on the user's specified POI type. The POIs present in these rows are transmitted back to the client application.
7. The client application decodes and sorts the results, and the nearest POIs are shown on her phone's local map.

4.3. Grid-based Cloaking

In our proposed protocol, the client application extracts the user's location via cell towers, Wi-Fi, or GPS, and it calculates the user's cloaking area by using the military grid reference system (MGRS) technique. MGRS is a geo-coordinate standard for locating points on the Earth [36]. The Earth is divided into grid squares with sizes of 0.1 km, 1 km, 10 km, 100 km, etc., based on the level of accuracy and degree of precision. Our proposed protocol uses MGRS to help ensure the user's location privacy. Each MGRS block is considered as a block in the database that is categorized based on POI type.

POIs are considered to be nearby if they are located in the same MGRS block as the user. The user's location privacy level increases if she chooses a larger MGRS block. However, a larger MGRS block includes more POIs, and it affects the computational cost of our proposed protocol. We will discuss this issue in more detail in the following section. Figure 2 shows the different levels of MGRS blocks for the Ottawa, Ontario, area [37].

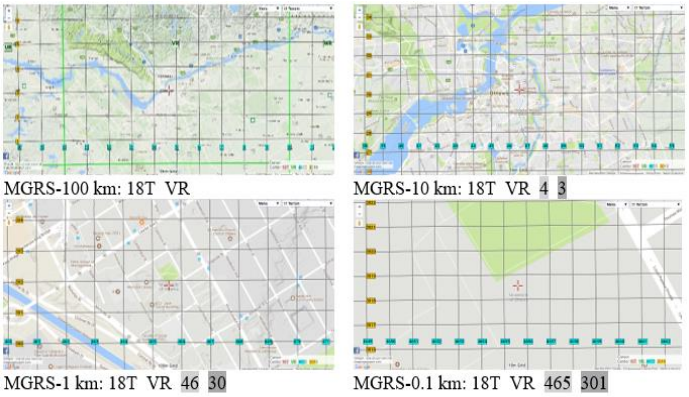
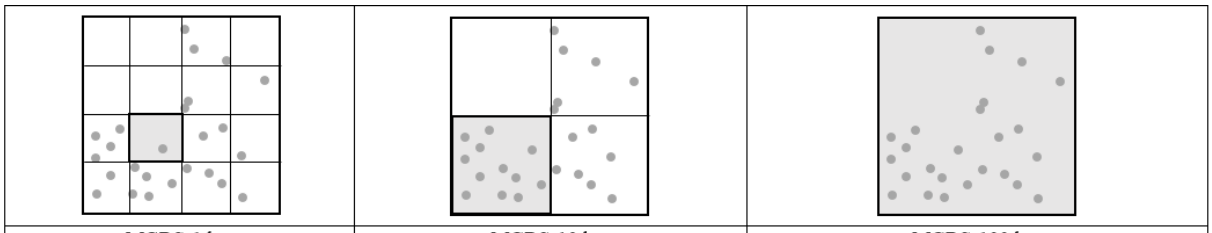


Figure 2 Different MGRS Levels [37]



MGRS-1 km

MGRS-0.1 km	POI type	POIs
18TVR 46 30	0000	null
	0001	null
	0010	POI1
	0011	null
	0100	null
	0101	null
	0110	null
	0111	null
	1000	null
	1001	null

MGRS-10 km

MGRS-1 km	POI type	POIs		
18TVR 4 3	0000	null	null	null
	0001	null	null	null
	0010	POI1	POI2	null
	0011	POI1	null	**null**
	0100	POI1	null	null
	0101	POI1	POI2	POI3
	0110	POI1	null	null
	0111	POI1	null	null
	1000	null	null	null
	1001	POI1	POI2	POI3

MGRS-100 km

MGRS-10 km	POI type	POIs			
18TVR	0000	POI1	POI2	null	null
	0001	POI1	POI2	POI3	null
	0010	POI1	POI2	null	null
	0011	POI1	null	null	null
	0100	POI1	POI2	POI3	null
	0101	POI1	POI2	POI3	null
	0110	POI1	null	null	null
	0111	POI1	POI2	POI3	POI4
	1000	POI1	POI2	null	null
	1001	POI1	POI2	POI3	POI4

Figure 3 Illustration of the relationship between the MGRS block, POI types, and POIs as saved in the LBS database.

4.4. Location Cloaking

The first step in the location cloaking phase is to apply different MGRS levels on the geographic region, such as Canada and the U.S.. Then, the user's cloaking area is calculated based on the user's selected MGRS level and her current location or the location of interest. POIs are considered to be nearby if they are located in the same cloaking area.

The selected MGRS level must be large enough to achieve the privacy of the user's location within the requested cloaking area, but simultaneously it must be small enough to reduce the computational overhead on the smartphone application to process the results, and also to reduce the communication overhead to transfer the result via the wireless data traffic.

In order to map POIs to a cloaking area, [9] used the Various-size-grid Hilbert Curve (VHC) technique. [9] chose VHC because it could solve the issue of density of POIs based on the geographic region. For example, the data traffic cost increases if the selected area has a high density of POIs (within a city). On the other hand, if the selected area has a lower density of POIs (within a countryside), then the result size decreases and the server is able to estimate the location of the user, which leads to a loss of privacy. VCH can solve this problem by creating a different-sized cloaking area based on the density of POIs. However, this solution has the disadvantage of receiving a list of POIs which may or may not be useful for the user. If the selected area has no POI that the user is looking for, she still has to wait for the client application to calculate the result, which is based on all POIs in a selected VCH region, and then show the result which is actually "null". This can cause a high computational cost on the client side application.

To manage the computational cost on the smartphone application based on the density of the POIs, and prevent high computational cost in the lower density POIs area, we propose a new technique to map POIs to a cloaking area based on the MGRS fix-sized blocks. First of all, we categorize POIs based on their types in each MGRS block. In Figure 3 we consider ten POIs types per MGRS block (see Appendix A) [7]. Then, each row of the database refers to the MGRS block concatenated with the POI type. As [9] mentioned, the density of POIs varies by geographic area. Therefore, each row of the defined database has a variable size.

To protect the privacy of the user's location and prevent the server from guessing which POI type is fetched by the user, we need to equalize the number of POIs in each selected cloaking area. Therefore, if the number of POIs in one POI is not the same as the maximum POI size in the selected cloaking area, the rest of the row must be set to "null". By this technique, our PIR client side computational cost relies on the location of the user and the level of selected MGRS. If the user's location has low POI density, the PIR client side computation time will decrease. If the user's location has high POI density, the PIR computational cost on the client side will increase. Note that the server cannot observe the differences between computational costs for queries in different locations, because we equalized the number of POIs in the selected cloaking area. Otherwise, the server which is able to observe the difference between computational cost based on different user's queries, could guess the user's location. Figure 3 shows an example of the POI density in the selected area based on different levels of MGRS and illustrates the relationships among the MGRS block, POI types, and POIs as saved in the LBS database.

5. Security Analysis of the Proposed Protocol

The user can use her current location or a location that she wishes to visit in the future. This feature adds one more level of privacy in our protocol because the observer or the location-based service provider (LBSP) is not aware of whether the requested MGRS block corresponds to the user's current location. Therefore, our proposed protocol has two kinds of privacy: first, protection of the user's location privacy within the requested MGRS block, and second, the LBSP or an observer does not know whether the

request is the user's current location at the request's time. In both cases, our main goal is to protect the user's location privacy against the LBSP and any other observer.

5.1. Threat model

We can apply our proposed PIR protocol to all existing block-based PIR schemes (CPIR and IT-PIR). In this section, we use the IT-PIR (multi-server) to describe our threat model (as was done by [9]). The primary assumption in IT-PIR schemes is that servers must not communicate with each other to breach the privacy of users' queries. Under this assumption, the IT-PIR protocol itself has been proven secure in [10, 15, 27]. Given the cryptographic security of the IT-PIR scheme, we review the security of our PIR protocol, as well as its security against passive and active adversaries in the following sections.

5.2. Security Analysis

Claim 5.2.1. If B is an MGRS block of level L, chosen by Alice, and T is the type of POI that Alice is searching for, our proposed privacy-preserving protocol for LBS is secure against a *passive adversary* in the block B.

Passive adversary. An external observer or a malicious LBSP who has access to the data exchanged between the user and the database along the communication channel but cannot change the data.

Justification. Our proposed protocol calculates the portion of the database in which Alice wants to find a nearby place of interest. It depends on the level of privacy she selected, such as MGRS-100 m. Therefore, Alice's location privacy and query are limited to the requested portion of the database. The number of POIs in each type of an MGRS block is set to be equal to the maximum number of POIs in that MGRS block by adding "null" to the ends of the other types in our proposed database, Thus, the passive adversary cannot guess which type of POI was fetched by Alice. In other words, if Alice's type of POI changes while she is still in the same MGRS block, then for a new request, Alice will send the same query for the same MGRS block of the database.

Because Alice selected her level of privacy to be, for example, a block of MGRS-100 m, it is impossible for the passive adversary to detect her movement as long as she is in the same MGRS block. Therefore, a correlation attack is actually unachievable since Alice will always send the same query for a given privacy level. Additionally, if Alice knows that she is going to move, she should choose a larger MGRS block that includes both her current location and her next location. Thus, her movement will not be detectable.

The PIR scheme provides the security for our proposed privacy-preserving protocol for LBS against a passive adversary. The only information which the passive adversary can access is the identity of the user. If the user's identity protection is required, we can use TOR. Additionally, the user's query contents and the response of the database can be protected against a malicious observer by applying end-to-end encryption techniques, for example, TLS (transport layer security) through the wireless communication channel. Both schemes (TOR and TLS) are optional for the user, as they slow down the protocol and cause an additional computational cost.

Claim 5.2.2. If B is an MGRS block of level L chosen by Alice, and T is the type of POI that Alice is searching for, our proposed privacy-preserving protocol for LBS is secure against an *active adversary* in the block B.

Active adversary. A malicious external observer who has access to data exchanged between the user and the database along the communication channel and can delete, insert or modify data. The LBSP is considered to be trusted in Claim 5.2.2.

Justification. Message reordering attack. During this attack, the active adversary attempts to delay and/or reorder requests or responses to mix up results and the communication. Note that in our proposed protocol, if the results of multiple queries are not received in proper sequence, it has no effect on the server or Alice since each result holds the requested information. The adversary also gains no information about Alice's query or location.

Justification. Message tampering attack. Alice will not be able to verify false responses if the adversary starts to send them. Therefore, a DoS (denial-of-service) attack is possible. However, in using this attack, the active adversary will not learn any more information about Alice's query or location, which is the main focus here. Note that this attack can be prevented by using TLS over the communication channel.

Justification. Message insertion or deletion attack. If an active adversary tries to delete or insert data from the server's response or Alice's request, it can cause a DoS attack. The adversary does not receive any information about Alice's request or location. Again, TLS can be used on the communication channel to prevent this attack, if desired.

Justification. Message replay attack. In this attack, if an active adversary starts a replay attack against Alice or the server, it does not affect either of them. The server responds to the requests, and Alice can easily drop multiple responses with the same information. The adversary will not receive any information about Alice's query or location. As with the previous attacks, we can use TLS to prevent this attack as well.

Claim 5.2.3. If B is an MGRS block of level L chosen by Alice, and T is the type of POI that Alice is searching for, our proposed privacy-preserving protocol for LBS is secure against a *malicious server* in the block B.

Malicious Server. A malicious server refers to an LBSP that attempts to insert new messages, modify, or delete messages in response to the user.

Justification. If the malicious server sends a false response, does not return a response, or sends additional messages with the

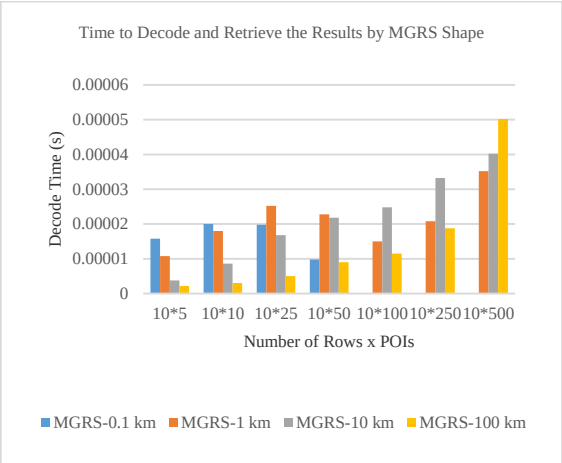


Figure 4 Comparison of time to decode and retrieve the results, by MGRS shape at the client side. It shows the computation time for queries on one MGRS block (ten POI types) for different number of POIs (each POI consists of 300 bytes).

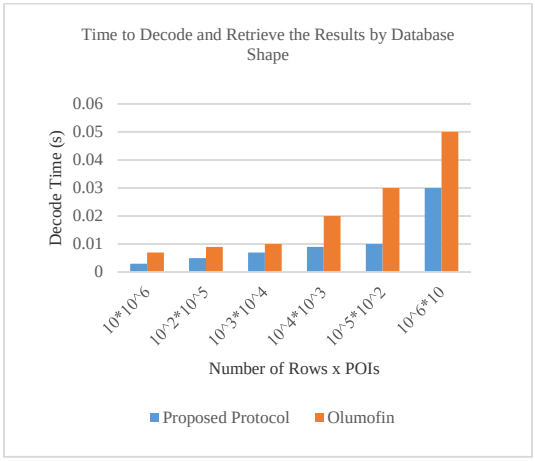


Figure 5 Comparison of decode and result retrieval time by database shape at client side in our proposed protocol and in Olumofin. The results show the computation cost for different-shaped database for queries on a 3 GB database (each POI consists of 300 bytes).

response, it can cause a DoS attack against the user. However, this attack does not enable the malicious server to learn any information about the location of the user or her query. The only thing that could help the malicious server find information about the location of the user is the content of the query, which is protected by using the PIR scheme within the MGRS block.

6. Experimental evaluation and results

We implemented a prototype of our proposed protocol based on an open source PIR protocol called Percy++ [15, 16, 38, 39, 40, 41]. We ran our C++ prototype on a virtual machine with Ubuntu Linux operating system and an Intel® Core™ i7-8550U CPU @ 1.80 GHz, 4GB RAM. We followed all assumptions of [9] in our implementation to compare our results with their approach. We randomly generated and distributed ten million POIs within Canada and the U.S. [7]. Each POI consisted of 300 bytes that included the longitude and latitude coordinates, name, exact address, phone number, website address, etc., of the POI [9]. We set the number of the databases to two to use the Percy++ PIR [9]. For each of these, we applied MGRS to our generated map to create four databases for four levels of user privacy: MGRS-0.1 km, MGRS-1 km, MGRS-10 km, MGRS-100 km.

Recall that the main goal of this section is to decrease the decode time on a smartphone to make the PIR scheme practical for resource-constrained hardware. The decoding time has a direct correlation with the number of POIs in each MGRS block. As we defined the fixed number of POI types for different levels of MGRS blocks (ten types), the PIR decoding time on the smartphone depends on the number of POIs in each type. Therefore, if the number of POIs in a type increases, the decoding time increases. We equalized the number of POIs in all types of MGRS blocks by adding "null" entries, as explained above. Therefore, there can be types that have no data or less than the maximum POI type. This improves the decoding time compared to a type that has the maximum number of POI. As observed in Figure 4 , when the user's level of privacy is increased (larger MGRS block), the probability of obtaining more POIs per type increases. Therefore, an MGRS block with a maximum of five POIs per type has faster data retrieval compared to 500 POIs per type. However, there are some exceptions. For example, MGRS-0.1 m required less computational time with 50 POIs per type than with five POIs per type. These exceptions can happen if the type that was requested has fewer POIs than 50, and the rest of the data in that type is "null" (to make the total number of POIs equal to the maximum number of POIs in the requested MGRS block). During the decode time, whenever the first "null" block is detected, the decode operation stops and results are returned to the user.

We ran our prototype using different levels of MGRS blocks, and we generated 100 random requests to calculate the average time to decode and retrieve the results. As observed in Figure 4, the probability that we had 500 POIs from one type in an MGRS-0.1 km was equal to zero. This means, for example, that in an area of 0.1 km^2, there cannot be 500 banks. The opposite scenario may also occur. The probability that we have five POIs of one type in an MGRS-100 km block is rare. Considering this, the MGRS-10 km is the best choice if we want to show the results for an MGRS block with various numbers of POIs/types.

In our proposed protocol, the number of rows in each MGRS block is the same as the number of POI types, which was set to ten types per MGRS block, regardless of the size of the MGRS block. Thus, if our MGRS block becomes larger, the number of types (rows) does not change, but the number of POIs per type increases. We also have different numbers of POIs per type. Thus, in an MGRS block, we have some types that have no data or less than the maximum number of POIs in that MGRS block. This is important because we want to compare our results with [9], and due to the different definition of privacy and the reasons we just mentioned, it is difficult to give an exact comparison.

However, to show the performance of our proposed protocol compared to [9], we followed their implementation by setting the privacy level equal to one. Thus, we considered one large MGRS block that covered Canada and the U.S., and the number of POI types was set equal to the number of rows in [9]. Our implementation results are based on a database of ten million POIs. Figure 5 shows the time for decoding and retrieving the results for various numbers of rows and POIs.

As observed in Figure 5, our performance is approximately 50% better than that of [9] because our method considers the POI type and uses the MGRS which applies a fix-sized cloaking area and a variable-sized block to the database. In our proposed protocol, the user receives exactly the type of POI that she was looking for. By increasing the number of POIs per row, the decode and retrieve time increases in [9] protocol because after decoding the rows of the database, the results must be filtered to show the POI that the user was looking for.

As stated in section 4.3, due to the four levels of MGRS (0.1 km, 1 km, 10 km and 100 km), our database required four different configurations. This was the only disadvantage of our proposed protocol compared to [9]. This could increase processing on the server when adding or removing POIs from the databases (for example, when a restaurant closes or a new one opens in a specific MGRS block).

7. Limitations of our Proposed Protocol

In general, there may be a case in which the user will not find a reasonable POI in the requested cloaking area. Therefore, she may wish to search further in a larger MGRS block (i.e., in a broader geographical area). When this happens, the user's privacy does not decrease in our proposed protocol; it is still guaranteed to the level of the original cloaking area.

As mentioned in section 4.3.3, due to the four levels of MGRS (100 km, 10 km, 1 km and 0.1 km), we required four different configurations for our database. This is a disadvantage of our proposed protocol compared to [9]. This could increase processing on the server when adding or removing POIs from the databases (for example, when a restaurant closes or a new one opens in a specific MGRS block).

Modern smartphones with multi-core processors may be able to handle the 1.5 GB database for Canada and the U.S. that is used in [9] evaluation section, as well as the most recent 3 GB location database [7] that we used in our implementation. However, we should mention that not all people have the most recent smartphones and so our proposal, which reduces computational cost on the client by almost 50%, may be of particular interest for such environments.

8. Conclusion and Future Work

In this paper, we presented a privacy-preserving protocol to help the user search for nearby places of interest while protecting her location's privacy by using PIR. For this purpose, we first proposed a block-based PIR scheme to decrease the computational overhead on smartphone applications [8]. We demonstrated that by applying our PIR scheme to the LBS, the computational overhead on the client side was reduced by approximately 50% compared to that reported in a previous work [9]. This reduction is valuable for the implementation of PIR in smartphone applications with limited resources. We demonstrated that our proposed LBS protocol is secure against active and passive attacks, as well as against a malicious server that tries to identify information about the user's query and location.

Our approach of retrieving the specific POIs in a cloaking area consumes less computational cost compared with the naive approach of asking the user to download the entire contents of the cloaking area and extract POIs locally. Simultaneously, the user's location privacy is not compromised since the user requests the same cloaking area as if she was requesting the entire contents of the cloaking area. This is a great benefit that reduces the cost of the wireless communication and also the memory usage on the smartphone.

There exist a number of interesting directions for our LBS privacy future work. First, our implementation results are based on [15]; in order to improve the computation, cost of our proposed protocol, we could develop it based on the higher performance block-based PIR such as the hybrid PIR which is proposed in [13]. Second, our proposed protocol could be extended by supporting more complex types of queries. Third, our proposed protocol could be combined with the Vehicle-to-Infrastructure (V2I) and the Vehicle-to-Vehicle (V2V) communication to help the user find her nearby places while she is driving her car. In this scenario, the user receives the latest update about the nearby places for her response from other vehicles or street infrastructure such as traffic lights and signs instead of a solid database. To update information about places and their availability constantly, we could use a Blockchain infrastructure in which other vehicles or street infrastructure are able to update their recent observations about the places. For example, all the infrastructure components on the street are connected to the Blockchain and every change that occurs appears in the block. Now if the user is looking for nearby parking, the traffic light could let her know the nearest parking lot and if there is any spot available by checking the updated list on the Blockchain.

References

[1] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: Anonymizers are not necessary" In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 121–132, 2008. https://doi.org/10.1145/1376616.1376631

[2] W. Hao, and Y-C. Hu, "Location Privacy with Randomness Consistency" In Proceedings on Privacy Enhancing Technologies, 62–82, 2016. https://doi.org/10.1515/popets-2016-0029

[3] U. Hengartner, "Hiding location information from location-based services" In International Conference onMobile Data Management, 268–272, 2007. https://doi.org/10.1109/MDM.2007.56

[4] M. Hezaveh, and C. Adams, "Privacy Preserving Discovery of Nearby-Friends" In E-Technologies: Embracing the Internet of Things, Lecture

Notes in Business Information Processing, (289), 41–55, 2017. https://doi.org/10.1007/978-3-319-59041-7_3

[5] A. Khoshgozaran, C. Shahabi, and H. Shirani-Mehr, "Location privacy: going beyond K-anonymity, cloaking and anonymizers" Knowledge and Information Systems, 26(3), 435–465, 2011. https://doi.org/10.1007/s10115-010-0286-z

[6] Y, G. Kim, J. Kong, and S. W. Chung, "A Survey on Recent OS-level Energy Management Techniques for Mobile Processing Units" IEEE Transactions on Parallel and Distributed Systems, 2388–2401, 2018. https://doi.org/10.1109/TPDS.2018.2822683

[7] Factual, Global Places-Schema, https://my.factual.com/data/t/places, last accessed 2018/11/05.

[8] M. Hezaveh and C. Adams, "A PIR scheme to improve the computation cost on the client-side of smartphone application" In IEEE 31th Canadian Conference on Electrical and Computer Engineering, 1–4, 2018. https://doi.org/10.1109/CCECE.2018.8447708

[9] F. Olumofin, P. K. Tysowski, I. Goldberg, U. Hangartner, "Achieving efficient query privacy for location-based services" International Symposium on Privacy Enhancing Technologies Symposium, 93–110 2010. https://doi.org/10.1007/978-3-642-14527-8_6

[10] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval" In Proceedings of the 36th Annual Symposium on the Foundations of Computer Science, 41–50, 1995. . https://doi.org/10.1145/293347.293350

[11] D. Lin, E. Bertino, R. Cheng, and S. Prabhakar, "Position transformation: a location privacy protection method for moving objects" In Proceedings of the SIGSPATIAL ACM GIS 2008 Inter- national Workshop on Security and Privacy in GIS and LBS, 62–71, 2008. https://doi.org/10.1145/1503402.1503414

[12] D. Riboni, L. Pareschi, and C. Bettini, "Privacy in georeferenced context-aware services: A survey" Privacy in Location-Based Applications, 151–172, 2009. https://doi.org/10.1007/978-3-642-03511-1_7

[13] C. Devet and I. Goldberg, "The best of both worlds: Combining information-theoretic and computational PIR for communication efficiency" In International Symposium on Privacy Enhancing Technologies Symposium, 63–82, 2014. https://doi.org/10.1007/978-3-319-08506-7_4

[14] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router" In Proceedings of the 13th conference on USENIX Security Symposium, 2004. https://svn.torproject.org/svn/projects/design-paper/tor-design.pdf

[15] I. Goldberg, "Improving the robustness of private information retrieval" In Proceedings of the IEEE Symposium on Security and Privacy, 131–148, 2007. https://doi.org/10.1109/SP.2007.23

[16] C. Aguilar-Melchor and P. Gaborit, "A lattice-based computationally-efficient private information retrieval protocol" In Western European Workshop on Research in Cryptology, 2007. https://eprint.iacr.org/2007/446.pdf

[17] R. Sion, and B. Carbunar, "On the computational practicality of private information retrieval" In Proceedings of the Network and Distributed Systems Security Symposium, 2007. https://zxr.io/research/sion2007pir.pdf

[18] W. Gasarch, "A survey on private information retrieval" The Bulletin of the EATCS, 82, 72–107, 2004. https://www.researchgate.net/profile/William_Gasarch/publication/266280304_A_survey_on_private_information_retrieval/links/5705098d08ae74a08e270e57.pdf

[19] B. Chor and N. Gilboa, "Computationally private information retrieval" In Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, 304–313, 1997. https://doi.org/ 10.1145/293347.293350

[20] B. Chor, N. Gilboa, M. Naor, "Private Information Retrieval by Keywords" Technion-IIT, Department of Computer Science, 1997.

[21] E. Kushilevitz, and R. Ostrovsky, "Replication is not needed: single database, computationally-private information retrieval" In Proceedings of the 38th Annual Symposium on Foundations of Computer Science, 364–373, 1997. https://doi.org/10.1109/SFCS.1997.646125

[22] C. Cachin, S. Micali, and M. Stadler, "Computationally private information retrieval with polylog communication" In International Conference on the Theory and Applications of Cryptographic Techniques, 402–414, 1999. https://doi.org/10.1145/258533.258609

[23] C. Gentry, and Z. Ramzan, "Single-database private information retrieval with constant communication rate" International Colloquium on Automata, Languages, and Programming, 803–815, 2005. https://doi.org/10.1007/11523468_65

[24] E. Kushilevitz, and R. Ostrovsky, "One-way trapdoor permutations are sufficient for non-trivial singleserver private information retrieval" In International Conference on the Theory and Applications of Cryptographic Techniques, , 104–121, 2000. https://doi.org/10.1007/3-540-45539-6_9

[25] H. Lipmaa, "An Oblivious Transfer Protocol with Log-Squared Communication" In International Conference on Information Security, 314–328, 2005. https://doi.org/10.1007/11556992_23

[26] C. Aguilar-Melchor, J. Barrier, L. Fousse, and M.O. Killijian, "XPIR: Private information retrieval for everyone" In Proceedings on Privacy Enhancing Technologies, 155–174, 2016. https://doi.org/10.1515/popets-2016-0010

[27] A. Beimel and Y. Stahl, "Robust information-theoretic private information retrieval" Journal of Cryptology, 20(3), 295–321, 2007. https://doi.org/10.1007/s00145-007-0424-2

[28] Y. Gertner, S. Goldwasser, T. Malkin, "A Random Server Model for Private Information Retrieval" In 2nd International Workshop on Randomization and Approximation Techniques in Computer Science, 200–217, 1998. https://doi.org/10.1007/3-540-49543-6_17

[29] S. Wang, X. Ding, R. H. Deng, and F. Bao, F, "Private information retrieval using trusted hardware" European Symposium on Research in Computer Security, 49–64, 2006. https://doi.org/10.1007/11863908_4

[30] E. Fung, G. Kellaris, and D. Papadias, "Combining Differential Privacy and PIR for Efficient Strong Location Privacy" International Symposium on Spatial and Temporal Databases, 295–312, 2015. https://doi.org/10.1007/978-3-319-22363-6_16

[31] S. Papadopoulos, S. Bakiras, and D. Papadias, "Nearest neighbor search with strong location privacy" In Proceedings of the VLDB Endowment, 3(1-2), 619–629, 2010. https://doi.org/10.14778/1920841.1920920

[32] G. Ghinita, "Understanding the privacy-efficiency trade-off in location based queries" In Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS, 1–5, 2008. https://doi.org/10.1145/1503402.1503404

[33] P. Paillier, "Public key cryptosystems based on composite degree residue classes" In International Conference on the Theory and Applications of Cryptographic Techniques, 223–238, 1999. https://doi.org/10.1007/3-540-48910-X_16

[34] A. Shamir, "How to Share a Secret" Communications of the ACM, 22(11), 612–613 1979. https://doi.org/10.1145/359168.359176

[35] A. Pingley, W. Yu, N. Zhang, X. Fu, and W. Zhao, "CAP: A Context-Aware Privacy Protection System For Location-Based Services" In 29th IEEE International Conference on Distributed Computin Systems, 49–57, 2009. https://doi.org/10.1109/ICDCS.2009.62

[36] S. Hemisphere, "Northern Hemisphere" Ann Arbor 1001, 2006. (see also: https://en.wikipedia.org/wiki/Military_Grid_Reference_System)

[37] Mapping support, https://mappingsupport.com/p/gmap4.php?tilt=off&mgrs=14SPG34308382&z=5&t=t1, last accessed 2018/11/05.

[38] C. Devet, I. Goldberg, and N. Heninger, "Optimally Robust Private Information Retrieval" In USENIX Security Symposium, 269–283, 2012. https://eprint.iacr.org/2012/083.pdf

[39] I. Goldberg. Percy++ project on SourceForge, http://percy.sourceforge.net/. last accessed 2018/11/05.

[40] R. Henry, F. Olumofin, I. Goldberg, "Practical PIR for Electronic Commerce" In Proceedings of the 18th ACM conference on Computer and communications security, 677–690, 2011. https://doi.org/10.1145/2046707.2046784

[41] W. Lueks, I. Goldberg, "Sublinear Scaling for Multi-Client Private Information Retrieval" In International Conference on Financial Cryptography and Data Security, 168–186, 2015. https://doi.org/10.1007/978-3-662-47854-7_10

Appendix A

A global, open, collaborative, standardized points of interest database provided by Factual [7].

	POI Type	Canada	US
1	Automotive	41913	617664
	Automotive, maintenance and repair	27267	285077
	Automotive, maintenance and repair, tires	0	73587
	Automotive, car parts and accessories	6917	79292
	Automotive, car dealers and leasing, car dealers	7729	105690
	Automotive, car dealers and leasing, used car	0	74018
2	Businesses and services, financial	35914	348814
	Businesses and services, financial, bank and finance, bank and credit union	14886	128993
	Businesses and services, financial, financial planning and investments	7919	91282
	Businesses and services, financial, access and bookkeeping	13109	128539
3	Businesses and services	93421	1181684
	Businesses and services, personal care, beauty salons and barbers	26148	325314
	Businesses and services, shipping freight, and material transportation	6232	51966
	Businesses and services, insurance	13687	224441
	Businesses and services, legal, attorney and law offices	12438	226907
	Businesses and services, real estate, real estate agents	9869	140201
	Businesses and services, Telecommunication services	6667	47031
	Businesses and services, computers	9684	92043
	Businesses and services, printing, copying and signage	8696	73781
4	Businesses and services, home improvement	138297	1403076
	Businesses and services, home improvement	93762	962116
	Businesses and services, home improvement, contractors	25612	260638
	Businesses and services, home improvement, ventilating and air conditioning, heating	5970	74885
	Businesses and services, home improvement, plumbing	6279	58550
	Businesses and services, home improvement, electrician	6674	46887
5	Community and government	57185	839554
	Community and government, organization and associations	14688	136389
	Community and government, education and secondary schools Primary and secondary school	14907	192245
	Community and government, day care and preschools	6859	80547
	Community and government, religious, churches	14531	362889
	Community and government, public and social services	6200	67484
6	Healthcare	51359	969517
	Healthcare, dentists	18683	230012
	Healthcare, pharmacies	11840	62652
	Healthcare, hospitals, clinics and medical centers	8221	155416
	Healthcare, physicians	12615	521437
7	Retail	81534	758818
	Retail, furniture and décor	8656	96082
	Retail, fashion, shoes	5946	61335
	Retail, fashion, clothing and accessories	22886	193354
	Retail, fashion, jewelry and watches	6115	67001
	Retail, construction supplies	5776	42307
	Retail, supermarkets and groceries	10862	98231
	Retail, food and beverage, beer, wine and spirits	6014	54942
	Retail, convenience stores	9610	100149
	Retail, glasses	5669	45417
8	Social	179478	1911585
	Social, food and dining, restaurants	101714	1017499
	Social, food and dinning, restaurants, fast food	17207	236356
	Social, food and dinning, restaurants, dining	16033	0
	Social, food and dinning, cafes, coffee and tea houses	15083	117367
	Social, food and dinning, restaurants, pizza	12888	136018
	Social, food and dinning, restaurants, American	0	225245
	Social, food and dinning, restaurants, Chinese	5724	0
	Social, Bars	10829	179100
9	Transportation	17906	206107
	Transportation, gas stations	11783	161466
	Transportation, taxi and car services, car and truck rentals	6123	44641
10	Travel	27937	177682
	Travel, travel agents and tour operators	6505	35613
	Travel, lodging, hotel and motels	21432	142069