PrAd: Enabling Privacy-Aware Location based Advertising

Hung Dang, Ee-Chien Chang
School of Computing
National University of Singapore
{hungdang,changec}@comp.nus.edu.sg

Abstract

Smart phones and mobile devices have become more and more ubiquitous recently. This ubiquity gives chance for mobile advertising, especially location-based advertising, to develop into a very promising market. In many location-based advertising services, it is implied that service providers would obtain actual locations of users in order to serve relevant advertisements which are near users' current locations. However, this practice has raised a significant privacy concern as various private information of an user can be inferred based on her locations and trajectories. In this work, we propose PrAd, a location-based advertising model that appreciates users' location privacy; i.e. it never reveals their locations to any untrusted party. Our solution is conceptualized based on several state-of-the-art privacy preserving techniques such as *data obfuscation*, *space encoding* and *private information retrieval* (PIR). We especially introduce algorithmic modification to existing hardware-based PIR technique to make it more practical and thus suit real-time applications. Moreover, PrAd enables a correct billing mechanism among involved parties without revealing any individual sensitive information. Finally, we confirm the effectiveness of our proposed framework by evaluating its performance using a real world dataset.

Categories and Subject Descriptors

H.2.8 [**DATABASE MANAGEMENT**]: Database Applications—*Spatial databases and GIS*

General Terms

Design, Security

Keywords

Privacy-Preserving, Location-based Advertising

1. INTRODUCTION

A tremendous growth in smartphone usage has been witnessed in recent years and it is expected that more than one-third of global population will use smartphones within the next few years. In such a context, mobile-advertising has become a promising market. One of the most popular forms of mobile-advertising is *location-based advertising* (LBA). This is a form of advertising that leverages location-based services to conduct mobile advertising. LBA offers a mechanism in which location-specific advertisements (those that are particularly relevant to a specific location) are delivered to appropriate consumers (those that are close to the advertisement's location, for example). For brevity, we simply refer to location-specific advertisements as *ads*. Most of today's LBA services track users' personal and private information in order to be able to serve the most relevant ads to the users. Such tracking has raised many concerns about privacy violation. To a certain extent, *location-based advertising server* (LBAS) has to take into consideration at least the locations of users and ads. However, LBAS should not be trusted as it may reveal users' locations to a third party without users' consent. Several studies have shown that location disclosing has great implications in term of privacy [1,2]. Given a location information of individuals, a broad set of other sensitive information such as health status or religious view could be inferred [1]. These concerns raise a need of protecting location privacy in LBA.

In this paper, we focus on a specific type of LBA that displays appropriate location-based ads on user's phone during their usage of ad-sponsored applications. We target the highest level of location privacy, which completely protects users' location privacy from any untrusted parties. The most prominent and powerful candidate of these untrusted parties is LBAS since it has access to users' location information.

Various techniques have been proposed to protect location privacy in the context of location based services (LBS) [3–8]. The main idea of these techniques is to enable location-based queries such as nearest neighbor queries or range query in such a way that actual locations of query points are not revealed. Among these solutions, there are three main categories. The first category [3,4] employs *location obfuscation* idea, in which a query point is either cloaked in a group of some other query points or blurred in an area so that the exact location of the query point cannot be inferred. However, in this class of techniques, user's location can be restricted in a small area of the space, thus it is relative easy to infer her location. The other category [5,6] uses *space encoding* techniques to hide actual location of Point of Interests (POI). Approaches based on this concept are still not able to protect users' location information from inference attacks where the untrusted LBAS may match users' queries with outliers or popular locations based on the access frequencies. The third major concept employs *private information retrieval* (PIR) [1,7,8]. Generally, PIR-based approaches utilize PIR protocol [9] to implement a query procedure in

GeoPrivacy'15, November 03-06, 2015, Bellevue, WA, USA
Copyright (c) 2015 ACM. ISBN 978-1-4503-3969-8/15/11$15.00
DOI: http://dx.doi.org/10.1145/2830834.2830839.

which database item is retrieved privately from location-based service without it learning which block was retrieved. Though this technique is resistant to attacks that the two other classes are vulnerable to, it has its own limitations. Approach proposed in [7] leaks the cardinality of the PIR request while scheme presented in [1] incurs a prohibitive computational cost. On a different perspective, there are many studies on privacy issues in mobile-advertising [10,11]. However, these studies focus on content personalization and targeted advertising instead of location-based advertising. Thus, they do not try to protect location privacy of users.

In this work, by adopting *space encoding* and PIR techniques, we propose PrAd, a novel LBA model targeting location privacy; i.e. enable location-based advertising without compromising location privacy of users. Our key insight is instead of sending location information to LBAS and letting it select appropriate ads to deliver to users, PrAd keeps the sensitive information on user's phone, carries out the selection locally, and then privately requests pertinent ads from LBAS. LBAS no longer obtains location information of users or processes spatial queries, which means it is deprived of sensitive information. Moreover, we design PrAd such that ads retrieval and delivery are carried out privately, i.e. without LBAS knowing which are requested and retrieved. We introduce three main privacy metrics and justify that PrAd satisfies all those three metrics, and thus can put the claim that PrAd can obtain location privacy in LBA.

PrAd encodes locations of both users and advertisements using one-way space encoding function. Mobile devices request ads by first computing its own location index, and sending such index to LBAS to retrieve pertinent advertisements. Only encrypted indices are sent out of the device and hence no location information ever leaves the device. The space encoding technique is designed so that locality and neighborhood of spatial objects are preserved. In PrAd approach, ads selection is performed locally on user's devices instead of being processed by LBAS as in traditional model. Utilizing this technique alone can protect users' location privacy in snapshots. That is, given a single advertisement retrieval, LBAS cannot find out from where the request is made, i.e. where the user currently is. However, based on access frequencies of records in its database, history queries and external geographic knowledge, LBAS can carry out inference or correlation attacks to deduce user's location. We further improve PrAd by adopting an ORAM-based *private information retrieval* technique [12] to completely nullify inference and correlation attacks. The PIR scheme proposed in [12] requires periodical offline processing to reshuffle the entire database, which implies not only a considerable computational and operational cost but also a degradation of quality of service due to the offline period. We introduce modifications to the scheme to avoid such reshuffling. We remark that by avoiding such shuffling, the service can operate continuously and smoothly without any interruption, which significantly enhance quality of service. Finally, PrAd uses homomorphic encryption techniques to ensure proper and accurate accounting as well as billing among LBAS, application publisher and advertisers. The only assumption PrAd makes is the presence of a piece of trusted and tamper proof primitive, namely *Secure Coprocessor* (SC), connect to LBAS's system in such a way that every request to LBAS is replayed through this SC.

2. BACKGROUND

In this section, we first present a generic view of LBA. We later discuss some background concepts in location privacy and provide a brief overview of PIR. Similar to other location privacy schemes [1,5,7], our goal is to protect user's location and identity information. In order to achieve these privacy measures, we place trust in trusted and tamper-proof unit residing on the untrusted server, receiving queries from users and privately retrieving appropriate records to answer such queries. The assumption of such a trusted primitive is indeed valid and has been employed in several works [7,13]

2.1 Location Based Advertising

Location-based advertising is a relatively new model compared to other types of advertising. LBA can be considered as a combination of *mobile advertising*, which is a form of advertising via mobile phones or devices, and *location-based service*, which is a class of services whose features are significantly controlled by location data. The basic principle behind LBA is to use technology to locate consumers' position and use that information to serve them with appropriate location-specific advertisements on their mobile phones or devices. Location-based services can generally be classified into two types. In *push* approach, service providers target and offer services to consumers without any specific request from the latter whereas users explicitly request for service in *pull* approach. LBA's main categories comprises of *messaging*, *display*, *search* and *product placement*.

In this work, we focus on a very popular scheme of LBA which leverages on *push location-based service* to offer *display mobile advertising*. This model involves several parties serving different roles. For simplicity, we informally define the four main parties participating in the service:

1. *Advertiser*: This party are businesses and marketers that want to advertise their products to customers. This party pays LBAS to get their ads delivered to the customer. LBAS, in turn, pays Application Publisher to display ads that it collected from Advertisers on their apps. We refer to such mobile applications as ads-sponsored or ads-funded apps.

2. *LBAS*: This party collects ads from Advertisers and then delivers them to mobile applications of customers in the ads' proximity.

3. *Application Publisher*: This party mostly includes mobile developers who distribute mobile applications to users free of charge and then gain benefit from Advertisers by displaying ads on their apps.

4. *User/Consumer*: This party represents advertisers' main target. Users install ad-sponsored applications on their mobile devices, which enables displays of ads on their devices during their usage.

Application Publishers who want to cooperate with LBAS include in their mobile applications a connection to LBAS. Using this connection, the mobile application can communicate with LBAS to fetch ads. LBAS should have access to both locations of ads and customers to realize LBA. This information obtaining is problematic from privacy perspective. We consider LBAS as an untrusted party and thus, user's location should not be revealed to LBAS. In this work, we propose a mechanism to offer LBA in such a way that the untrusted LBAS can learn nothing about user's location.

2.2 Location Privacy Preliminaries

Privacy Metrics. Assume that an untrusted LBAS hosts an ads database $ADB = \{a_1, a_2, a_3, ..., a_n\}$, in which a_i is a set of ads relevant to point of interest (POI) l_i and that a set of users U=$\{u_1, u_2, u_3,....,u_m\}$ subscribe to S's services, our target is to enable users to privately retrieve ads in such a way that no sensitive location or identity information is disclosed to the untrusted LBAS. We consider an ad-retrieval as a spatial query issued by the user and the answers for such queries are appropriate ads. We adopt privacy metrics defined in [7] in our work.

Definition 1 (u-anonymity). *Given a query, with respect to the server's knowledge, the user who issues the query should be indistinguishable among the entire set of users. That is, for every query q, the probability $P_q(u_j)$ that user u_j issues query q is the same for every user, i.e. $P_q(u_j) = \frac{1}{m}$ where m is the total number of users*

The above definition is to ensure the untruted LBAS is blinded from the information of who issues the query. In addition, we also need to hide the location from which the query is issued.

Definition 2 (a-anonymity). *The location at which the query is issued should be kept secret. That is, for every query q, with respect to the server's knowledge, the probability $P'_q(l)$ that the query q is issued at location l is the same for every location, i.e. $P'_q(l) = \frac{1}{area(A)}$ where A is the entire region covering all POIs.*

We argue that privacy measures implemented by the two above definitions are much stronger than metrics used in other *anonymity* approaches [3, 4, 14]. In such notions, a user is only indistinguishable among a small set of k-1 other users or her location is hidden in a small region R. In fact, the privacy requirements of Definitions 1 and 2 stimulate an extreme case of other *anonymity* approaches where $k = m$ (a user is indistinguishable among *all* users) and $R = A$ (user location is blurred into the *entire* region).

While *a-anonymity* and *u-anonymity* can guarantee the privacy of the query in snapshot, they still reveal the access frequency, which allows the untrusted server to carry out correlation attack [7]. To prevent this, LBAS should obtain no information about which item is retrieved from it per each request. Thus, we propose that a query should be evaluated in a data-oblivious way. We use the similar definition of data obliviousness as defined in [15]

Definition 3 (Data-oblivious Execution). *An execution is considered data-oblivious if it incurs the same sequences of operations and memory accesses for any two inputs of the same length.*

Thread model. The purpose of an adversary is to learn users' location. We assume the most powerful adversary, who pretends to be a normal user and together with the untrusted LBAS conspire against the user. Note that the LBAS can play an adversary by self-issuing queries and observing records' access pattern to find the correspondence between user's location and records hosted on it. In this work, we consider LBAS as a primary adversary.

2.3 Private Information Retrieval

In PIR setting, a database is modelled as a n-bit string $X = \{X_1, X_2, X_3, ..., X_n\}$ hosted on an untrusted server S, and the user is interested in retrieving the i^{th} bit in X, which is X_i, without revealing the value of i. A broad range of PIR schemes can be classified into cryptographic and hardware-based approaches.

Cryptographic PIR. The original PIR scheme is proposed in an information-theoretical setting in which even an adversary with infinite computational power cannot find out the value of i. However, it is proven that in theoretical PIR setting, the communication cost is equivalent to the size of the entire database. Thus, in order to reduce such an overhead, several computational PIR approaches only try to ensure that computationally bounded adversary cannot find i within polynomial time [16]. Even though they can mitigate the huge communication cost, they still have to perform a linear scan on the entire database. This class of PIR suffers from a prohibitive communication and computation cost, which makes it less practical in real applications.

Hardware-based PIR. In order to obtain strong privacy without suffering from high costs, a class of Hardware-based PIR has been proposed [12, 17, 18]. These approaches assume a tamper-resistant hardware device is installed on the untrusted server (which is the LBAS in our case). Such a device, in several cases referred to as *Secure Coprocessor (SC)*, is equipped with hardware cryptographic accelerators that are able to execute fast and efficiently cryptographic operations. Hardware-based PIR approaches trust the *SC* to privately perform information retrievals. By placing trust on the SC, these techniques achieve optimal communication and computation costs in comparison with cryptographic PIR approaches. Because of this, we employ this class of PIR approaches to build our privacy-aware *LBA* system.

2.4 Homomorphic Encryption

We also utilize basic additive homomorphic encryption to carry out the accounting. Additive homomorphic encryption system is an asymmetric cryptosystem that allows addition operation to be performed directly on ciphertexts. In details, each plaintext x is encrypted using a public key pk. Given a public key pk, anyone can calculate the sum of $E(pk, x_1)$ and $E(pk, x_2)$, to generate a result which is $E(pk, x_1 + x_2)$. This result, when decrypted with a secret key sk corresponding to pk, renders the plaintext x_1+x_2. Note that in performing an addition of $E(pk, x_1)$ and $E(pk, x_2)$ using pk, no information on x_1 and x_2 is revealed.

3. PRIVACY-AWARE LBA

As discussed above, with a tremendous growth of smartphone usage, LBA has become a very promising market. However, privacy concerns, especially location privacy, to a certain extent, discourage a portion of users to participate in this market. We argue that if LBA service does not compromise users location privacy, it will be much more broadly accepted and its market will be further extended.

We make a simple yet elegant observation that instead of disclosing the location information of consumers to LBAS as in traditional model, a privacy-aware LBA can keep such information locally on user's smartphone and privately requests pertinent ads from LBAS. In specific, mobile devices perform simple computation to find out which ads should be served in current spatial context and then *privately* retrieves those ads from LBAS, without the latter knowing which ads are actually delivered to the consumers. Later on, ads reports, which tell how many times certain ads are displayed, are collected anonymously to ensure no sensitive information is leaked. By guaranteeing privacy in each phase of the LBA serving, we can protect users' location privacy. PrAd, a framework for location-privacy aware LBA, implements exactly the above mentioned observation.

3.1 Architecture

PrAd proposes two changes to the traditional LBA architecture. The first change involves the presence of a trusted primitive, namely Secure Coprocessor (SC) (figure 1). Specifically, SC is connected to LBAS and every request to LBAS is routed through it. Moreover, it is presumed that SC is able to read and write data from and to LBAS's storage and that no other process or adversary can tamper with or inspect the internal state of SC; i.e. it is inaccessible to the LBAS. Such primitive can be implemented using IBM secure cryptographic processors [19] or the recently proposed Software Guard Extensions (SGX) from Intel [20].

The second component of PrAd is a small service called *mPrAd* running in user's device. Unlike traditional LBA where each ads-sponsored application individually fetches ads directly from LBAS, they are all bound to *mPrAd* to retrieve ads PrAd model. *mPrAd* first collects exact location information from sensor (i.e. GPS sensor) and performs the *space encoding* (see subsection 3.2) to get a corresponding location index i. It then establishes a secure channel to SC and submits the request via that secure channel. The SC receives the request, performs a *private ads retrieval* (detailed in subsection 3.3). It then returns the ads to *mPrad*. Upon receiving response from SC, *mPrAd* forwards them to the requesting applications. Every time an advertisement is displayed on the application's GUI, it sends an acknowledgement to *mPrAd*. Based on these acknowledgements, *mPrAd* keeps track of how many times each advertisement is displayed. At the end of a billing period, *mPrAd* constructs billing vectors and reports them anonymously to the LBAS (subsection 3.4). We sequentially discuss details of each procedure in the following subsections.

mPrAd contains two secret keys, one for performing space encoding and the other to encrypt billing vector. The reason to delegate these tasks to a small independent service like *mPrAd* is that ads-sponsored applications are not always trusted. There are incentives for such applications to leak users' current location to the untrusted parties. Hence, ads-sponsored application should not be granted access to location sensor, which is required in the first place for space-encoding process, unless there is an explicit need of location information to facilitate its authorized activities. The other reason is that there are so many apps developers that distributing a set of secret keys required in our model incurs many complications. By introducing *mPrAd*, we can avoid such secret key distribution problem. One more reason is that there may be several ads-funded applications running in the same smartphone; and running a single *mPrAd* to serve all of those ads can save computational and communication cost compared to each application perform ads retrieval separately.

3.2 Space Encoding

Unlike traditional LBS model in which an user sends her location information l to the server and let the server queries for which service is pertinent to l and then serves the user with those services, the client device in our model makes such a selection on its own. To realise this, we assume that LBAS stores ads in a database $\mathcal{D} = \langle A_1, A_2, \ldots, A_n \rangle$, in which A_i is a set of advertisements co-located at location l whose index is i. The client first calculate an index i from her location l, and then requests A_i from LBAS. The process of computing i from l is referred to as *space encoding*.

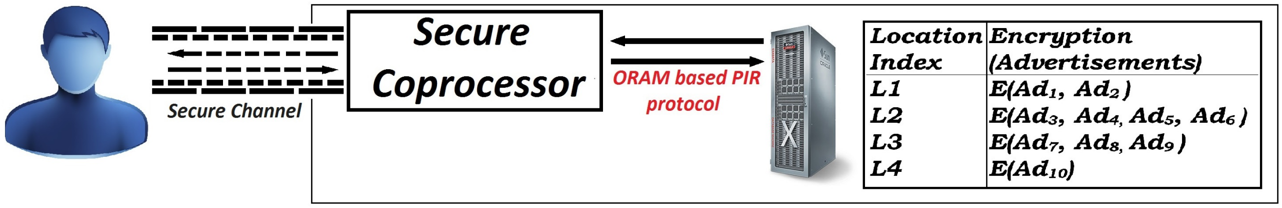
Since we want to protect user's location against the LBAS, the space encoding has to be a one-way function where forward computation (compute i from l) is easy to performed while invert computation (inferring l from i) is computationally impossible. Moreover, because the ads selection process needs real-time performance and that smartphone has limited computational power, the space encoding's computational cost should be low. Besides, to ensure the correctness of the advertising service, the space encoding must preserve the locality and clustering aspects of spatial data. Given these requirements, we adopt a Hilbert curve based space-encoding technique proposed in [5] as a space encoder in our model. We briefly present the notion of *space filling curve* and summarize the technique in [5].

A space filling curve visits all points in the space without crossing itself. This class of curves retains the proximity and locality aspects of spatial data. Hibert curve [21] is one of the most famous member of this class due to its excellence in preserving distance and clustering characteristics of the spatial data. The space encoding technique introduced in [5] uses a set of four parameters to form a *Space Decryption Key SDK*. The four parameters comprise of the curve's starting point (X_0, Y_0), curve orientation O, curve order N and curve scale factor F. It is proven that this space encoder is secure, i.e. it is computationally impossible to invert the encryption without the knowledge of $SDK = \langle (X_0, Y_0), O, N, F \rangle$. Given ads are stored based on location index is calculated by this space encoding and all ads contents are encrypted properly, LBAS has no information to locate the user when it serves her request.

In the real world, many ads are located in the same area. For example, there are many products from different brands offered in one supermarket, and many supermarkets co-located in the same neighborhood. PrAd groups all ads that are in the same area into one set and then indexes such a set with the location index of that area. Note that the size of such an area is a sensitive parameter in our framework. We discuss the effect of this parameter in our evaluation.

Without loss of generality, let us presume that there is a minimum bounding rectangle surrounding the entire region. We divide such a rectangle into x unit squares, each of which represents an area. Note that there will be areas (unit squares) containing ads and some others don't. PrAd only keeps track of areas that contain ads. Besides all ads submitted by advertisers, PrAd stores a special ads record which is to be served to users having no ads in their proximity.

Our framework allows users to flexibly decide a range of the area in which they want to retrieve ads. *mPrAd* treats a wide area as an union of several unit squares. In case a



Location Index	Encryption (Advertisements)
L1	E(Ad_1, Ad_2)
L2	E(Ad_3, Ad_4, Ad_5, Ad_6)
L3	E(Ad_7, Ad_8, Ad_9)
L4	E(Ad_{10})

Figure 1: Private ads retrieval

user prefers to retrieve ads in an area comprising of many unit squares, *mPrAd* only needs to calculate location index of its current location. Based on this, it infers indices of surrounded unit squares and requests the corresponding advertisements from LBAS.

3.3 Private Ads Retrieval

Because the space-encoding is a one-way function, LBAS cannot invert the transformation to infer actual location of the users from their submitted indices. Hence, it can protect user's location privacy in snapshots; i.e. for a single request, user's location information is kept private. However, her privacy is still vulnerable to correlation attack. In such, the LBAS, with external geographic knowledge of the area, observes the access frequencies and history trajectories of the user to infer her locations. PrAd takes one step further to address this issue. All communication between the SC and LBAS storage is performed obliviously; i.e. without the latter learning the intended access of the former.

In specific, PrAd aims to enable SC to privately retrieve selected ads from the LBAS such that LBAS cannot learn which ads SC truly wants to get. Moreover, we also want to support update operation. When an advertiser join or leave the system, the set of advertisement records captured in LBAS's database changes. Thus, to offer flexibility for PrAd, supporting update operations is necessary. Due to privacy concern, the LBAS must be kept oblivious to what operation SC performs on its database. In another word, LBAS does not know whether an access that SC made is a read or a write or which item the SC intentionally accesses.

To realize this, PrAd adopts an idea of hardware-based PIR techniques due to its optimal communication and computation costs. It is worth mentioning that placing a trust on a secure coprocessor (SC) connected to LBAS is fundamentally different from trusting the LBAS. We only need to trust the SC's designer, instead of paying credits not only to the LBAS's designer but also its administrator and all other applications installed on it. Another observation is that it is easier to vet the SC compared to screening the LBAS.

Several SC-based PIR protocols employ *random permutation* techniques to first permute the original database DB into permuted one (DB_p) and later on access DB_p to privately retrieve records [12, 22, 23]. Even though these approaches are able to obtain optimal communication and computation costs, they need to carry out an *offline preprocessing* to reshuffle the entire database periodically. The cost of this offline reshuffling is not trivial and thus make these protocols inapplicable in our context. We introduce some modifications to avoid these overheads. As the result, our private ads retrieval is able to achieve optimal communication and almost optimal computation cost without periodically performing the reshuffling. The key insight is instead of performing one big reshuffle periodically, we slightly change the database after each retrieval. Such a change should be minimum to keep the processing cost low, yet still significant enough to nullify LBAS's correlation and access pattern attacks. In the following, we present a protocol that allows SC to privately access LBAS's database. SC issues *Access* (*Read*, A_i, *null*) to retrieve A_i or *Access* (*Write*, A_i, v) to update A_i with new value v.

The architecture. An ads database $\mathcal{D} = \langle A_1, A_2, \ldots, A_n \rangle$ consisting of n records, is hosted on LBAS. SC is connected to LBAS and able to read and write records from and to the LBAS's database. SC comprises of a private memory M with a limited storage capacity. As SC is tamper-resistant, it follows that its private memory is also trusted and LBAS cannot have any access to or observation on the content of the memory. Every request to LBAS is configured to be routed through SC. The role of the SC is to serve as a proxy sitting between users and LBAS.

The Oblivious Protocol:

```
Access($op, A_i, v$):
  $x \leftarrow pos[A_i]$
  $y \leftarrow flag[x]$                          ▷ if x = null, y = null
  if $x = null$ then
      $a \leftarrow PickWhite()$
      $b \leftarrow PickBlack()$
  else if $y = white$ then
      $a \leftarrow x$
      $b \leftarrow PickBlack()$
  else                                            ▷ $x \neq null$ & $y = black$
      $a \leftarrow PickWhite()$
      $b \leftarrow x$
  end if

  $ReadCell(a)$, $ReadCell(b)$
  $value \leftarrow A_i.value$
  if $op = Write$ then
      $A_i.value \leftarrow v$
  end if

  $A_m \leftarrow PickRandomS()$
  $A_n \leftarrow PickRandomS()$
  $WriteCell(a, A_m)$, $WriteCell(b, A_n)$

  $pos[A_m] \leftarrow a$, $pos[A_n] \leftarrow b$
  $pos[A_a] \leftarrow null$, $pos[A_b] \leftarrow null$
  $flag[a] \leftarrow black$, $flag[b] \leftarrow black$

  return $value$
```

In the *LBAS initialization* or *pre-deploy* phase, the SC randomly chooses $n - k$ records, encrypts them using a semantically secure encryption, in which re-encryption of the same value and encryptions of different values are indistinguishable, and then writes them to the permuted

DB_p, stored on LBAS's storage server. For the sake of exposition, we consider this storage server comprises of $n-k$ cells, each of which is allocated to one record. Other k records are cached in SC's memory, which will keep track of the following three data structures:

1. *stash* S: a list of k records
2. *pos*: a map keeping track of positions of $n-k$ records stored in the LBAS storage.
3. *flag*: a boolean array to color cells on LBAS storage. $black$ cells are those that are already accessed; $white$ cells are those that have not been accessed yet.

The above protocol makes use of the following supporting functions: $PickWhite()$ and $PickBlack()$ are to choose a white and black cell uniformly from random, respectively; $PickRandomS()$ arbitrarily selects one item among k records currently located in the stash S; $ReadCell(x)$ reads an advertisement record located at cell x on the LBAS storage into S whereas $WriteCell(x, A_i)$ writes a records A_i from S to cell x on LBAS storage.

For an access to an advertisement record A_i, the protocol first checks the state of A_i. If it is in S, the protocol accesses two random cells, one *white* and one *black*; otherwise, it accesses the cell containing A_i and another random cell of alternative color. After reading two cells from external storage to S, SC chooses two random items in S to write back to the accessed cells, mark these cells as *black*, and update the position mapping accordingly. Once all cells are marked as *black*, the SC flips the *flag* array to mark all except one random cell as *white* and start the session all over again.

3.4 Billing and Accounting

Client apps need to report to LBAS a number of times each advertisement is clicked or displayed for appropriate billing and accounting. We refer to this feedback as *ads-report*. It is necessary to hide the knowledge of what advertisements displayed on which users' mobile apps in other to guarantee users' location privacy. Thus, ads reports should be collected in an anonymous way. Instead of each individual sending her ads report directly to LBAS and advertiser, a group of users first aggregate their ads report and only the aggregated information is sent to LBAS and advertiser. These pieces of accumulated ads reports will not reveal sensitive information of individual users but are sufficient to perform billing and accounting.

We take advantages of homomorphic encryption and *k-anonymity* [24] to maintain the anonymity of ads reports. In detail, for every advertiser, $mPrAd$ keeps track of a number of times its ads are displayed. This information is captured in the form of $\langle Advertiser - Counter\rangle$. Within one billing period, whenever a counter for a specific advertiser reaches a predefined threshold, say C, $mPrAd$ encrypts the counter using homomorphic encryption to get encrypted value EC, put it to a message of form $\langle Advertiser - P - EC\rangle$ where P is a number of extra peers the message has to be routed through before reaching LBAS and sends it out to another peer. Initially, P is set to some threshold MP. Upon receiving a message, the peer checks whether its own counter for that advertiser is greater than 0. If it is, it updates EC by first encrypting its own counter and then homomorphically adding its own counter to EC. It later resets the counter to 0, decrease P by one. It later sends EC to LBAS if $P = 0$, or forwards the accumulated billing message to a next peer otherwise. In case the receiving peer's counter for the advertiser specified in the message is 0, it proceeds with the same procedure except for updating EC. At the end of billing period, client app encrypts and sends out every counters which are greater than 0.

Since the time constraint in reporting billing information is quite flexible, these billing message could be transmitted using Delay-Tolerant Network (DTN) [25] to save energy and communication cost. Note that the selection of receiving peer is performed randomly using DTN technique, which increases the anonymity of the accumulated ads reports. In other to prevent one peer from inferring which ads are displayed on the previous peer, $mPrAd$ can pair each ads report which an empty ads report, i.e. ads report whose counter is 0. Upon receiving billing counters of advertisers, LBAS decrypts them and update its billing database accordingly. Billing information can also be sent to advertisers so that they can verify the correctness of LBAS billing.

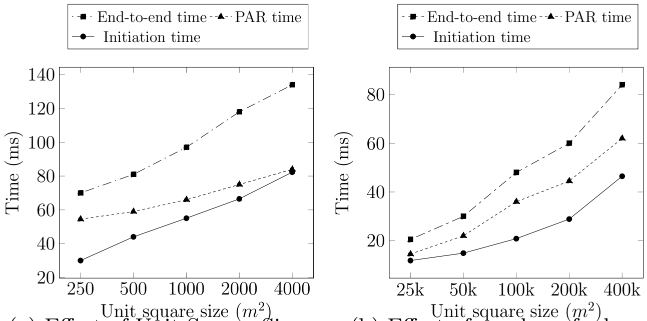
3.5 Security

PrAd satisfies all three security metrics discussed in section 2, and thus offers strong location privacy. In addition, the use of homomorphic encryption and k-anonymity concept in PrAd 's billing procedure further fortifies our claim.

Given the presences of SC, there is no direct interaction between users and LBAS. The only interaction between LBAS and SC is facilitated through the private protocol described in subsection 3.3. That is, in the view of the LBAS, every request is exactly the same as each other no matter by whom it is issued. Thus, our technique satisfy the *u-anonymity* metric. Thanks to the space encoding as well as the randomized encryption and database permutation privately performed in each ads retrieval, LBAS cannot figure out which POI is actually received interest. That is, it does not know from where user sent their request. Thus, *a-anonymity* is guaranteed. Finally, with respect to the view of LBAS, the SC performs exactly the same sequence of operations for access. Specifically, it reads one *black* and one *white* records from LBAS, and then replaces those two records with randomly chosen and encrypted item from SC's stash. We contend that our ads retrieval approach is *data-oblivious* and nullifies LBAS's ability of inferring any sensitive information based on database's access frequency. It is clear that our technique appreciates all three privacy metrics, *a-anonymity*, *u-anonymity*, and *data-oblivious execution*. In billing process, ads-reports are collected anonymously so that LBAS cannot learn which ads is displayed on which user's device. At this point, we claim that PrAd achieves strong location privacy in providing LBA service.

4. EVALUATION

We empirically evaluate the overall effectiveness of PrAd with respect to its scalability and the effect of unit square's size on its performance. We perform these sets of experiment on an Intel Core i7-2600 processor with 8GB of memory. In order to emulate a secure coprocessor, we limit our CPU clock to one tenth of its original power and use only 128MB of RAM to represents SC's cache. These choices are based on the *IBM 4765 Cryptographic Coprocessor* [19]. Our experiments are performed using YELP



(a) Effect of Unit Square Size (b) Effect of number of ads

Figure 2: Experiment results on PrAd performance

dataset [1], which comprises of 42,153 businesses, 252,898 users and 31,617 check-in sets covering large cities such as Phoenix, Las Vegas, Madison, Waterloo and Edinburgh. We consider businesses as advertisers, YELP users as clients who use ads-sponsored applications in our model. We treat a check-in data as a user visiting advertisers' locations; i.e. she will retrieve ads near those advertisers' locations. As one advertisers can launch several ads, we will consider a number of reviews a YELP business gets as a number of ads its representing advertisers offer. The intuition behind this is that the more popular the YELP business is, the more reviews it gets, which is analogous to the fact that the more dominant the advertiser is, the more ads it offers. We stimulate ads content as a string of 100 characters.

In our two sets of experiments, we report three metrics which are the pre-deployed LBAS initialization time, SC processing time (PAR time) and overall ads retrieval time on client's phone (end-to-end time). The first metric is reported as an average value of a hundred attempts while the other two metrics are averaged over a thousand ads retrieval requests. As the experiments reported, all these three metrics are in the order of milliseconds, which suggests that PrAd performance satisfies real-time requirement and thus practical deployment.

4.1 The effect of unit square size

In our model, ads are grouped and indexed with respect to a unit square. The size of these unit square is a very sensitive parameter. If it is too large, there may be so many ads grouped into one record, which may leads to rendering unrelevant ads to the clients. The other disadvantage is that as the cost of encryption and decryption directly depends on the size of a record, if the record is too big, the overhead will be high. On the other hand, if the unit square is too small, PrAd ends up performing several ads retrieval for each request, which is again incurs overhead.

In this set of experiment, we fix the number of ads to 400k, and varies the size of the unit square from 250 to 4k square meters and report the three metrics. The result is reported in figure 2a

As we can observe from the figure, as the size of the unit square increases, all three metrics increase. The reason for these increases is that the larger the unit square is, the more ads are grouped into one unit. This in turn implies larger records. Recall that the dominant operations in our protocol are cryptographic operations whose costs are directly dependent on the size of the record, the increasing in record size leads to higher processing cost. Also note that as the sizes of each ads record increases, the delay in end-to-end time also further increases since more data needs to be transferred during each ads retrieval.

4.2 Scalability

In the second sets of experiments, we varies the number of ads that LBAS serves and report the same three metrics as in the first set of experiment. The experiments witness the increase of all three metrics. In specific, end-to-end time grows by almost 4 times when the number of ads increases from 25k to 400k, and so do PAR and initiation times (see Figure 2b). We remark that though all the three metrics are increasing when the number of ads that PrAd serves increases, the performance is still practical, as all the metrics are only in order of miliseconds. We believe that this is efficient enough to support real-time applications. Thus we claim that PrAd is scalable and practical.

5. RELATED WORK

Data Transformation. Inspired by cryptographic techniques, a number of works [5, 26, 27] have advocated for the use of data transformation in protecting location privacy. Khoshgozaran et al. [5] utilized a Hilbert-based transformation to support privacy-preserving evaluating of approximate Nearest Neighbor (NN) queries. Another work proposed by Wong et. al. [27] uses a different transformation which preserves relative distances among all points of interest in the spatial database. While the approach in [5] can only render approximate result, this technique enables answering accurate kNN queries. Later, Lien et al. [26] suggest using Moore curve and Paillier cryptosystem to perform a secret circular shifts of spatial data so that kNN queries can be processed in a privacy-preserving manner. However, this technique can only provide almost accurate results rather than exact answers. Note subtly that these techniques are deterministic; the same encoded results are always rendered for the same queries. Though they can protect users' privacy in snapshots, they are vulnerable to correlation attacks [7].

PIR-based Location Privacy. The PIR concept was originally introduced by Chor et al. [9] and has been extensively studied over years by both research and industry communities [12, 28–30]. Secure hardware PIR is the most practical mechanism among all PIR techniques. Khoshgozaran et al. [7] adopt this concept to protect location privacy in kNN queries. The main idea of this technique is to reduce a query processing to a set of PIR block retrieval executed by the trusted secure coprocessor. Though each block retrieval is completely private, the untrusted LBS is still able to infer user's location by observing a number of PIR request for each query. Ghinita et. al. [1] present another technique that can completely protect users' location privacy as each query incurs exactly one PIR request. However, this technique is limited to single NN queries and suffers from prohibitive performance overhead.

Private Advertising System.There are several works on designing privacy-aware advertising system to protect users' privacy. The first class of these systems target personalized online advertisement on ordinary browsers. Adnostic [31] and Privad [32] enable private advertising by maintaining users' profiles locally on their computers. The selection of ads shown on users' displays are performed based on these profiles. Juels [33] utilizes PIR and mix network to protect users' privacy. RePriv [34], from Microsoft, provides

[1] http://www.yelp.com/dataset_challenge

private advertising based on user's browsing behaviour.

Another class of private advertising system focuses on mobile advertising. MoRePriv [11] advocates for OS-level service to solve a conflict of privacy and content personalization on mobile devices. SmartAds [35] only reveals the ad keyword, while keeping all other sensitive information private. MobiAd [36] downloads and displays advertisement based on users' interest profiles. Despite providing a certain level of privacy, these schemes still leak some user information to the server. Moreover, they pay very little attention to location privacy issue. Hence, none of these works can be directly applied to LBA.

6. CONCLUSION

In this work, we have proposed PrAd, a privacy-preserving LBA framework that enables correct ads serving and billing. PrAd leverages on *space-encoding, private information retrieval* and *homomorphic encryption* to enable the LBAS to provide location-based advertising service without compromising users' location privacy. This privacy aware mechanism will arguably encourage more consumers to join the system and thus give rise for the market. We have intuitively proven the security of our system and evaluated its performance using real world data set. The experiments show that PrAd operates at scale and in real-time. With the realization of hardware-attested secure primitive such as those proposed in [19,20], we believe that the deployment of such privacy-preserving LBA model is realistic.

7. REFERENCES

[1] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: Anonymizers are not necessary," SIGMOD '08.

[2] G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu, "Differentially private spatial decompositions," ICDE '12.

[3] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *Knowledge and Data Engineering, IEEE Transactions on*, 2007.

[4] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: Query processing for location services without compromising privacy," VLDB '06.

[5] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," SSTD'07.

[6] M. L. Yiu, G. Ghinita, C. S. Jensen, and P. Kalnis, "Enabling search services on outsourced private spatial data," *VLDB'10.*

[7] A. Khoshgozaran, C. Shahabi, and H. Shirani-Mehr, "Location privacy: Going beyond k-anonymity, cloaking and anonymizers," *Knowl. Inf. Syst.*, 2011.

[8] S. Papadopoulos, S. Bakiras, and D. Papadias, "Nearest neighbor search with strong location privacy," VLDB'10.

[9] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," 1997.

[10] M. c and S. Nath, "Privacy-aware personalization for mobile advertising," CCS '12.

[11] D. Davidson and B. Livshits, "Morepriv: Mobile os support for application personalization and privacy," tech. rep., May 2012.

[12] S. Wang, X. Ding, R. H. Deng, and F. Bao, "Private information retrieval using trusted hardware," ESORICS'06.

[13] S. Bajaj and R. Sion, "Trusteddb: A trusted hardware-based database with privacy and data confidentiality," *Knowledge and Data Engineering, IEEE Transactions on*, 2014.

[14] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," MobiSys '03.

[15] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," *Journal of the ACM*, 1996.

[16] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: Single database, computationally-private information retrieval," FOCS '97.

[17] D. Asonov and J.-C. Freytag, "Almost optimal private information retrieval," PET'02.

[18] A. Iliev and S. Smith, "Private information storage with logarithmic-space secure hardware," I-NetSec 04, 2004.

[19] "Ibm 4764 pci-x cryptographic coprocessor." `http://www-03.ibm.com/security/cryptocards/pcixcc/overview.shtml`.

[20] "Software guard extensions programming reference." `https://software.intel.com/sites/default/files/managed/48/88/329298-002.pdf`.

[21] D. Hilbert *Mathematische Annalen*, 1891.

[22] X. Ding, Y. Yang, and R. Deng, "Database access pattern protection without full-shuffles," *Information Forensics and Security, IEEE Transactions on*, 2011.

[23] Y. Yang, X. Ding, R. H. Deng, and F. Bao, "An efficient pir construction using trusted hardware," ISC '08.

[24] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 2002.

[25] K. Fall, "A delay-tolerant network architecture for challenged internets," SIGCOMM '03, 2003.

[26] I.-T. Lien, Y.-H. Lin, J.-R. Shieh, and J.-L. Wu, "A novel privacy preserving location-based service protocol with secret circular shift for k-nn search," *Information Forensics and Security, IEEE Transactions on*, 2013.

[27] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," SIGMOD '09.

[28] R. Sion, "On the computational practicality of private information retrieval," in *NDSS'07*, 2007.

[29] J. Trostle and A. Parrish, "Efficient computationally private information retrieval from anonymity or trapdoor groups," ISC'10.

[30] F. Olumofin and I. Goldberg, "Revisiting the computational practicality of private information retrieval," FC'11, 2012.

[31] V. Toubiana, H. Nissenbaum, A. Narayanan, S. Barocas, and D. Boneh, "Adnostic: Privacy preserving targeted advertising," 2010.

[32] S. Guha, B. Cheng, and P. Francis, "Privad: Practical privacy in online advertising," NSDI'11.

[33] A. Juels, "Targeted advertising ... and privacy too," CT-RSA 2001.

[34] M. Fredrikson and B. Livshits, "Repriv: Re-envisioning in-browser privacy."

[35] S. Nath, F. X. Lin, J. Padhye, and L. Ravindranath, "Smartads: Bringing contextual ads to mobile apps."

[36] H. Haddadi, P. Hui, and I. Brown, "Mobiad: Private and scalable mobile advertising," in *MobiArch2010.*