PrAd: Towards enabling Privacy-Aware Location based Advertising

Hung Dang

Abstract

Smart phones and mobile devices have become so ubiquitous that mobile advertising, especially location-based advertising have developed into a very promising market. With many location-based advertising services, it is implied that service providers obtain actual locations of users so that relevant advertisements, which are near users' current locations, can be served to them. However, this practice has raised a significant privacy concern as various private information of a 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. never reveals their locations to any untrusted party. PRAD is conceptualized based on several state-of-the-art privacy preserving techniques such as *data obfuscation*, *space encoding* and *private information retrieval*. Moreover, PRAD enables a correct billing mechanism among involved parties without revealing any individual sensitive information. We confirm the effectiveness of our proposed framework by evaluating its performance using real-world datasets.

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 3 years [1]. 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 (who are near 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. Location disclosing has great implications in term of privacy [2, 3]. Given a location information of individuals, a broad set of other sensitive information such as health status or religious view could be inferred [2]. These concerns raise a need of protecting location privacy in location-based advertising.

In this paper, we focus on a specific type of LBA that displays an ads on user's phone during their usage of ad-sponsored applications when they are in the ads' proximity. We target the highest level of location privacy, which completely protects users' location privacy from any unstrusted parties. The most prominent and powerful candidate of these unstrusted 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) [4–11]. 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 [4–6] 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 [7–9] 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 unstrusted LBAS may match users' queries with outliers or populate locations based on the access frequencies/patterns. The third major concept employs *private information retrieval* (PIR) [2, 10, 11]. Generally, PIR-based approaches utilize PIR protocol [12] to implement a query procedure in 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 [10] leaks the cardinality of the PIR request while scheme presented in [2] incurs a prohibitive computational cost. On a different perspective, there are many studies on privacy issues in mobile-advertising [13–15]. 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 that instead of sending location information to LBAS and let it select appropriate ads to deliver to users, PRAD keeps the sensitive information on user's phone, carry out the selection locally, and then privately request pertinent ads from LBAS. LBAS no longer obtains location information of users or processes spatial queries, which means it is deprived from 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 user's location using one-way space encoding function to get a location index that represents the location. The only data sent out of the device is this encoded index; i.e. no location information ever leaves the device. The space encoding technique is designed so that locality and neighborhood of spatial objects are preserved. Given that location information of ads are also encoded by the same techniques, it is because of this very property that ads selection could be performed locally on user's devices instead of being processed by LBAS as in traditional model. The mobile application then requests ads with relevant location index from the LBAS. Utilizing this technique alone can protect users' location privacy in snapshots. That is, given a single ads 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 [16] to completely nullifies inference and correlation attacks. That is, PRAD leaves LBAS no opportunity to infer user's location. As the result, users are served with relevant ads while their locations are protected. 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-hardware called *Secure Coprocessor* (SC) installed on LBAS's system.

In summary, we claim the following contributions in this work:

- We propose PRAD, a framework enabling location-privacy aware LBA based on a set of state-of-the-art privacy preserving techniques.
- We justify how PRAD obtains the highest level of location-privacy in the context of LBA.
- We evaluate the performance of PRAD using real-world dataset to show the scalability and practicality of our proposed framework.

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 [2,7,10], our goal is to protect user's location and identity information. In order to achieve these privacy measures, we place trust in a Secure Coprocessor [17], which is a trust tamper-proof hardware residing on the untrusted server, receiving queries from users and privately retrieving appropriate records to answer such queries.

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*[1], which is a form of advertising via mobile phones or devices, and *location-based service*[2], 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 location 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, which are *push* and *pull*. In push approach, service providers target and offer services to consumers without any specific request from them. In other approach, users explicitly issue requests, by entering a keyword into a search box of a certain application for example, and services are only served upon those requests. Since LBA is also relevant to mobile advertising, we also list here its main categories; i.e. *messaging*, *display*, *search*, *product placement*.

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

- $Advertiser$: This party often comprises of businesses and marketers that want to advertise their products to customers who are within the products' proximity. Advertiser is the one who pays $LBAS$ to get their ads delivered to 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.
- $LBAS$: This party serves the role of a broker who collects ads from Advertisers and then delivers them to mobile applications of customers in the ads' proximity.
- *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.
- $User/Consumer$: This party represents advertisers' main target. Users install ad-sponsored applications on their mobile devices so that ads can be displayed 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. In order for the LBAS to be able to deliver ads to mobile applications of customers who are in the ads' proximity, it should have access to both locations of ads and customers. Specifically, LBAS obtains information on user's location, based on which it distributes relevant ads to user's phone. This information obtaining is problematic in 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 $ADB = \{ad_1, ad_2, ad_3, ..., ad_n\}$, in which a_i is a set of ads relevant to point of interest (POI) l_i and that a set of users

[1] $http : //en.wikipedia.org/wiki/Mobile\_advertising$

[2] $http : //en.wikipedia.org/wiki/Location - based\_service$

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 unstrusted LBAS. We consider an ads-retrieval as a spatial query issued by the user and the answers for such queries are appropriate ads. We adopt privacy metrics defined in [10] in our work.

DEFINITION 1 (U-ANONYMITY). *Given a query, 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 unstruted 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). *Given a query, the location at which the query is issued should be kept secret. That is, for every query q, 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 [4–6]. 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 [10]. To prevent this, LBAS should obtain no information about the item retrieved from it and rendered to the user. 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 [18]

DEFINITION 3 (DATA-OBLIVIOUS EXECUTION). *An execution is considered data-oblivious if it has the equivalent sequence of operations and memory accesses for any two inputs with the same running time.*

Thread model The purpose of an adversary is to locate where the user is when she is served with ads. We assume the most powerful adversary, who pretends to be a normal user and together with the unstrusted LBAS conspire against the user. Note that the LBAS can plays an adversary by self-issuing queries and observing records' access pattern to find the correspondence between user's location and records hosted on it.

2.3 Private Information Retrieval

In PIR setting, a database is modeled 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 retrieve 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 where it is assumed that 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 [19]. Even though they can mitigate the huge communication cost, they still have to perform a linear scan of the entire database. Despite being able to achieve perfect secrecy of the item retrieval, this class of PIR suffers from a prohibitive communication and computation cost, which makes its 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 [16, 20, 21]. These approaches assume that there is a tamper-resistant hardware device installed on the untrusted server (which is the LBAS in our case). Such a device, in several cases referred to as *Secure Corocessor (SC)*, is equipped with hardware cryptographic accelerators that are able to execute fast and efficient crytographic operations. Hardware-based PIR approaches trust the SC to privately perform information retrievals. By placing trust on the SC, these techniques are able to achieve optimal communication and computation costs in comparison with cryptographic PIR approaches. Because of this very advantage, we employ this class of PIR approaches to build our privacy-aware LBA system.

2.4 Homomorphic Encryption

In this work, 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 on ciphertexts and gives an encrypted result. This encrypted result is decrypted into a value which matches the result of an addition carried out on plaintexts. In details, each plaintext x is encrypted using a public key pk. Given a public key pk, anyone can calculate the sum of $E(sk, x_1)$ and $E(sk, x_2)$, to generate a result which is $E(sk, x_1 + x_2)$. This result, when decrypted with a secret key sk corresponding to pk, will render the plaintext $x_1 + x_2$. Note that in performing an addition of $E(sk, x_1)$ and $E(sk, x_2)$ using pk, no information on x_1 and x_2 is revealed.

3. PRIVACY-AWARE LBA

As discussed above, given a tremendous growth of smartphone usage, LBA has become a very promising market. However, privacy concern, especially location privacy concern, in a certain extend, discourages a portion of users to participate in this market. We argue that if LBA service doesn't compromise users location privacy, it will be much more broadly accepted and its market will be further extended. The key idea behind our privacy-aware LBA model is to keep location information locally on user's smartphone instead of disclosing it to LBAS as in traditional model. The mobile devices perform some simple computation to figure out which ads should be served in current spatial context and then $privately$ requests those ads from LBAS. Later on, ads reports, which tells how many times a certain ads is displayed, are collected in such a way that leaks no sensitive information. By guaranteeing privacy in each phase of the LBA serving process, we can protect users' location privacy. In this section, we propose PRAD, a framework for location-privacy aware LBA and discuss its underlying concepts.

3.1 Architecture

PrAd proposes two changes to the traditional LBA architecture. As displayed in figure 1, the first component of PrAd is a small service called $mPrAd$ running in user's smartphone. Unlike traditional model of LBA where ads-funded mobile applications establish connections and retrieve ads directly from LBAS, in our model, they just need to call $mPrAd$ service. Specifically, when an ads-funded application is in use, it is bound to $mPrAd$ to retrieve ads. $mPrAd$ first collects exact location information from sensor (i.e. GPS sensor) and perform the *space encoding* (subsection 3.2) to get a location index. It then establishes a secure channel to SC, encrypts the request containing the location index with SC's public key and submits it. The SC receives the encrypted request, decrypts it and perform a *private ads retrieval* (detailed in *Algorithm 1*). The SC then encrypts the ads with $mPrad$'s public key and sends it back to $mPrAd$. Upon receiving response from SC, $mPrAd$ decrypts the response and forward a set of ads to mobile applications. For every ads displayed, the application sends an acknowledgement to $mPrAd$. Based on the acknowledgement, $mPrAd$ can keep track of how many times each ads is displayed. At the end of a billing period, $mPrAd$ builds billing vectors and reports them anonymously to the LBAS(subsection 3.4). We sequentially discuss details of these procedures in the following subsections.

$mPrAd$ contains two secret key, 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-funded applications are not always trusted. There are incentives for such ads to leak user's sensitive information, which is user's location in this context. Mobile applications can be designed to look benevolent to pass vetting procedure of apps publisher but become malicious once it is installed on user's phone [22]. Hence, ads-funded 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 application 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.

The second change involves the presence of a trusted Secure Processor SC. Specifically, SC is required to be installed on LBAS and every request to LBAS is routed through this SC. Note that this SC is able to read and write data to LBAS's database.

3.2 Space Encoding

Unlike traditional LBS model in which the users send their location information to the server and let the server decide on which services they will be served, the client device in our model makes such a selection on its own. We consider the traditional mode of LBA as *Server assigning ads* while we classify our approach as *User selecting ads*. We assume that LBAS stores ads as a dictionary of form <*Location Index - Set of Ads*> where *Location Index (LI)* is calculated using a one-way function based on the ads' locations. Specifically, assume that a user is at location l, instead of asking the LBAS "please send me ads near l", the client calculates the location index li that represents l and then retrieves from LBAS ads whose index is li. That is, we change the ads query to the ads selection. We refer to the process of calculating location index as *space encoding*.

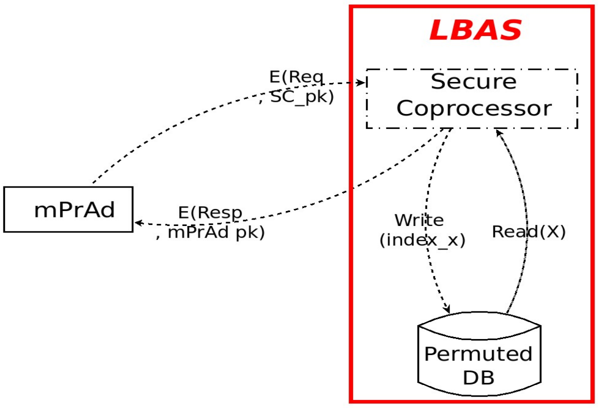

Figure 1: PrAd **architecture**

The space encoding needs designing so that it is a one-way transformation. A transformation is one-way if forward computation could be performed easily while backward or invert computation is computationally impossible. Moreover, since this space encoding is performed on users' mobile devices, its computational cost in term of time and computational resources should be low. The reason for this constraint is that the ads selection process needs real-time performing and that smartphone has limited computational power compared to a server. Another observation is that the basic nature of ads selection is to select not only an ads located exactly as user's location, but also ads in her proximity. Moreover, each user has different preference on a range of the proximity; i.e. some users prefer to retrieve ads within a small area surrounding them while other users opt to retrieve ads in a slightly larger area. Thus, the space encoding must preserve the locality and clustering aspects of spatial data. In order to satisfy the above mentioned requirements, we adopt a Hilbert curve based space-encoding technique proposed in [7] as a space encoder in our model. We briefly present the notion of *space filling curve* and summarize the technique.

Space filling curves is the one that visits all points in space without crossing itself. This class of curves retains the proximity and locality aspects of the spatial data. Hibert curves [23] 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 [7] 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 in the paper that this space encoder is secure, i.e. it is computationally impossible to invert the encryption without the knowledge of SDK. Given ads are indexed based on *Location Index* 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-worlds, 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. (This reasons our assumption on LBAS storing ads database as a dictionary of form <*Location Index - Set of Ads*>). Note that the size of such an area is a sensitive parameter in our framework. We discuss the effect of this parameter in subsection 4.1. Without loss of generality, let us presume that there is a minimum bounding rectangle that surrounds the entire region. Then, 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 space encoder utilizes space-filling curve, which reserves neighborhood and locality of spatial data object; i.e. if two spatial object are near each other, their spatial indices calculated by our space encoder will also very close to each other. Because of this very interesting property, our framework allows extra flexibility in serving ads. Specifically, users can decide on 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 user opts to be served ads in an area comprising of many unit square, let us say s for example, $mPrAd$ will send s requests to LBAS. $mPrAd$ only needs to calculate one location index of the user's current location; other location indices of surrounded unit squares are inferred based on the calculated on. For example, if the user opts to retrieve ads in an area that is 3 times larger than default unit square, and her current location's index is l, then $mPrAd$ will sends 3 requests with indices *l-1*, *l*, *l+1* to LBAS.

3.3 Private Ads Retrieval

So far, we have discussed how the idea of *User selecting ads* can be practical using space-transformation technique. Given this technique alone, user's location privacy is protected in snapshots. Without the knowledge of how the space-encoding calculate the location index, LBAS cannot invert the transformation and infer actual location of the users. Thus, for a single request, user's location information is kept private. However, her privacy is still suffered from correlation attack in which the LBAS observes the access frequencies and history trajectories of the user. It can make quite good inference on her locations based on those observations and external geographic knowledge of the area. In order to further improve the privacy guarantee and completely prevent any potential sensitive information leakage, we propose to utilize PIR technique in Ads retrieval. Our target is to enable client app to retrieve ads without the LBAS knowing what ads it has served.

The key challenge in our model is to privately retrieve selected ads from the LBAS. We adopt Hardware-based PIR techniques to obtain the privacy in ads retrieval due to its optimal communication and computation costs compared to cryptographic PIR. It is worth mentioning that placing a trust on a secure coprocessor (SC) installed on LBAS is fundamentally different from trusting the LBAS. In case of the SC, we only need to trust its designer, while the trust that we place on LBAS involve credits paid to not only its designer but also its administrator and all other applications installed on it. Another observation is that SC is a tamper-resistance hardware device which is programmed to perform some specific operations. 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 [16, 24, 25]. 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 the random permutation 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. We now present the SC-based Private Ads Retrieval protocol.

The architecture An ads database DB, which comprises 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.

Protocol In the *LBAS initialization* or *pre-deploy* phase, the SC encrypts $n - k$ records using a randomized encryption and write them to the permuted DB_p while keeping k remaining records in its memory (k is chosen according to SC's memory capacity and $n - k$ to be encrypted records are chosen randomly). Note that those k ads are cached in SC's memory in plain-text form; while other $n - k$ ads are saved in DB_p in their cipher-text form. Thus, one ads record will have two form, plain-ads if it is in SC's memory, or cipher-ads if it is in DB_p and it can only be of one out of two form at a time (since it is stored either in M or DB_p but not both). The randomized encryption generates different cipher-text each time the same plain-text is encrypted. If the encryption is not randomized, the access frequency of ads is partially leaked to the LBAS. In order to keep track of the permutation of n records in the original database DB, SC maintains the mapping table $MapT$ in its private memory M. Specifically, each item $MapT[li]$ is a pair of form $< a, x >$ where li is the location index of the original ads record, x is a position of the cipher-ads in DB_p and a is the boolean value signaling whether position x has been accessed. If an ads is stored in M (in plain-ads form), the corresponding a and x are set to -1 since its cipher-ads is not stored in DB_p. There are three possible values that an item $MapT[li]$ can take:

- *MapT[li] = (-1, -1)*: the ads is in plain-ads form and is kept in M
- *MapT[li] = (0, x)*: the ads is in cipher-ads form, kept in DB_p at position x. Position x has NOT been accessed before. This cipher-ads is marked as $unaccessed$ cipher-ads.
- *MapT[li] = (1, x)*: the ads is in cipher-ads form, kept in DB_p at position x. Position x has $ALREADY$ been accessed before. This cipher-ads marked as $accessed$ cipher-ads.

In the *online retrieval* phase, for each request, the SC will read from the DB_p one *unaccessed cipher-ads* and one *accessed cipher-ads* to M. Note that these cipher-ads are randomly chosen. It then randomly chooses two plain-ads from M, encrypts and write them to DB_p to replace the two recently read cipher-ads in DB_p; i.e. they are written to the exact position of the recently read cipher-ads. These two positions are then marked as $accessed$. $MapT$ is updated accordingly after each retrieval. After all cipher-ads are marked as $accessed$, SC randomly leaves one as $accessed$ and remarks all remaining *n-k-1* cipher-ads in DB_p as $unaccessed$ to start a new round. Details of the online retrieval process are given in *Algorithm 1*.

```
Retrieve(Location Index li)

if li not in $Index\_List$ then
    $li = Default\ Index$;
end
if MapT[li] = (-1,-1) then
    Let U be a random $unaccessed$ cipher-ads in $DB_p$;
    Let A be a random $accessed$ cipher-ads in $DB_p$;
else
    if MapT[li] = (0,x) then
        Let U be $DB_p[x]$;
        Let A be a random $accessed$ cipher-ads in $DB_p$;
    else
        Let U be a random $unaccessed$ cipher-ads in $DB_p$;
        Let A be $DB_p[x]$;
    end
end
u = Read(U);
a = Read(A);
Select in $M$ ads whose location index is $li$ and send it to client;
Write(u);
Write(a);
Update $MapT$ accordingly;
```

Algorithm 1: Online Retrieval

```
Read(cipher-ads C)

Read $C$ from $DB_p$ to $M$;
Remove $C$ from $DB_p$;
Decrypt $C$ to get its corresponding plain-ads $P$ and save $P$ in $M$;
Return a position of $C$ in $DB_p$;
```

Algorithm 2: Read

Note that there are regions in which there is no Advertisers publishing ads. To be able to deal with a situation in which users request ads from a non-ads area, SC keeps a list called $Index\_List$ which stores location indices of all ads records LBAS servers. Upon retrieving a request, it first checks if the requested location index is available. If it is not, SC will reassign the requested location index to a default value $Default\ Index$. Besides all ads published by Advertisers, LBAS keeps a special record whose index is *Default Index* and content includes some particular advertisements that LBAS serves for free (for example, these advertisement could be from a charity organization or an advertisement of LBAS's own service).

```
Write(position x)

Let $P$ be a plain-ads in $M$;
Encrypt $P$ to get is corresponding cipher-ads $C$;
Put $C$ at position $x$ in $DB_p$;
Remove $P$ from $M$;
```

Algorithm 3: Write

3.4 Billing and Accounting

In order to bill the advertisers, LBAS must know how many times a certain ads is displayed (or clicked). Client apps need to report to LBAS which ads is clicked or displayed so that LBAS can charge advertisers and pay application developer accordingly. We refer to this feedback as *ads-report*. However, we insist that it is necessary to hide the knowledge of what advertisements displayed on which users' mobile apps in other to guarantee users' location privacy. Thus, it is required that ads reports are collected in an anonymous way. Our intuition is that 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 won't reveal sensitive information of individual users but are sufficient to perform billing and accounting.

We take advantages of homomorphic encryption concept and *k-anonymity* [26] to maintain the anonymity of ads reports. In details, $mPrAd$ keeps counters of ads displayment of the form <*Advertiser - Ads Counter*>. Within one billing period, whenever a counter for a specific advertiser reaches a predefined threshold *maximum counter* MC, $mPrAd$ encrypts the counter using homomorphic encryption to get encrypted value EC, put it to a message of form <*Advertiser-P-EC*> where *P* is a number of peers the message has to be routed through before reaching LBAS and Advertiser and sends it to another peer. Initially, *P* is set to 0. Upon receiving a message, the peer checks whether its own counter for that advertiser is greater than 0. If it is, it encrypts the value to get an encrypted value EC_1, resets the counter to 0, and then performs the summation of EC and EC_1. It then updates EC value in the message to the encrypted result of the summation, increases *P* by one and check if *P* now reaches a predefined threshold MP (maximum peers). If *P* is now equal to MP, it sends EC to both LBAS and the corresponding advertiser. If *P* is still smaller than MP, it sends the accumulated billing message to a next peer. In case the receiving peer's counter for the advertiser specified in the message is 0, it just increases *P* by 1, and then based on the new value of *P*, it will sends the billing message to LBAS and corresponding advertiser or to the next peer. At the end of billing period, client app encrypts and sends out every counters which are greater than 0 even if they haven't reached MC yet. Since the time constraint in reporting billing information is quite flexible, these billing message transmitting could routed using Delay-Tolerant Network (DTN) [27] to save energy and communication cost. Note that the selection of receiving peer is performed randomly using DTN technique, which increases the anonimity 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 of an ads whose counter is 0. Upon receiving billing counters of advertisers, LBAS, having secret key of the homomorphic encryption, decrypts them and update its billing database accordingly. Note that because billing information is also sent to advertisers so that they can verify the correctness of LBAS billing.

3.5 Security

We claim that PrAd can obtain strong location privacy. We justify this claim by sequentially proving that in delivering ads to users, PrAd satisfies all three security metrics discussed in section 2. 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 retrieval detailed in *Algorithm 1*. 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 doesn't know from where user sent their request. Thus, *a-anonymity* is guaranteed. Finally, as described in *Algorithm 1*, with respect to the view of LBAS, the SC performs exactly the same sequence of operations and (permuted) database access for every ads retrieval. Specifically, it reads one accessed record and one unaccessed record to its memory M, and then replace those two records with randomly chosen and encrypted item from M. Thus, we argue that our ads retrieval approach is *data-oblivious*, which 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 leaks no sensitive information on users' actual location to untrusted party and hence, it 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* [28]. Although this coprocessor is very slow, it is equipped with cryptographic accelerators that offer a very efficient performance in performing cryptographic operations. Our experiments are performed using YELP dataset [3], 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 will 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 3 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.

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 3 metrics. The result is reported in figure 2

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 as a unit square becomes larger, there are more ads grouped into one unit, which makes the sizes of each ads records (a set of ads within one area) larger. Recall that the dominant operations in our protocol are cryptographic operations, whose costs are directly dependent on the size of the input, it is the reason for the increases of all processing times. Also note that as the sizes of each ads record increases, the delay in end-to-end time also further increases as more data needs to be transferred during each ads retrieval.

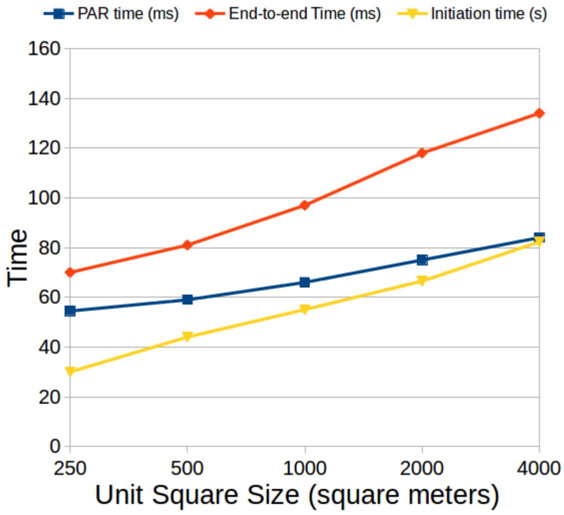

Figure 2: Effect of Unit Square Size on PrAd performance

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

4.2 Scalability

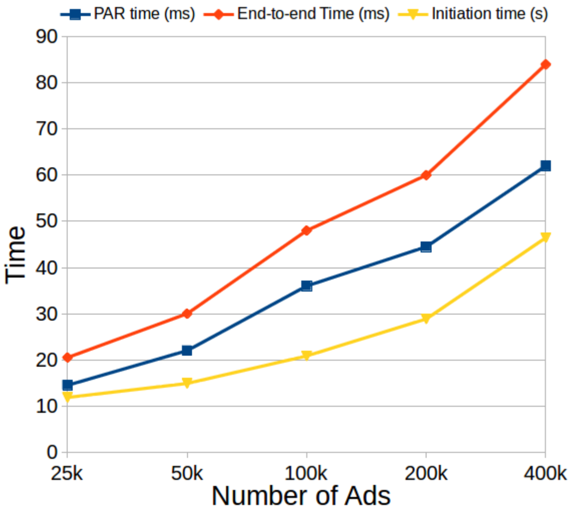

Figure 3: Effect of number of ads on PRAD performance

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. Figure 3 depicts the increases in all three metrics as the number of ads increases. This is very intuitive as the cost of cryptographic operations increases. The other reason is that as the number of ads records increases, the size of $MapT$ also increases, which partially affects the processing time of SC in each ads request. As the result of this set of experiments, it is clear that query response time is reasonable, which is just in order of miliseconds, and thus we claim that PRAD is scalable and practical.

5. RELATED WORK

Data Transformation. In this technique, data is encoded before uploading to the LBS, who is able to perform search and query processing on encoded data. Clients, having access to secret encoding key, issue encoded queries to LBS. Since both records in database and queries are encoded and unreadable to LBS, location privacy is obtained.

[7] utilized a Hilbert-based transformation to preprocess spatial data before storing it in the database. The transformation is considered as an encryption functions and its parameters serve the same role as encryption key. This technique can evaluate approximate Nearest Neighbor (NN) queries directly in transformed data points. Other work proposed by Wong et. al. [29] uses a different transformation which preserves relative distances of all point POIs in the spatial database. This approach enables the LBS to answer accurate kNN queries. [30] uses Moore curve and Paillier cryptosystem to perform a secret circular shifts of spatial data. The technique provides almost accurate answer for kNN queries.

Because of encryption-like properties, data transformation approaches can reasonably protect users' location privacy. However, they are still vulnerable to access pattern attacks as the same encoded results always are always rendered for the same queries. Specifically, if the LBS can observe that a particular ciphertext is returned many times, and it also has external geographical knowledge of popular POIs in the area, it can make inference on true locations of ciphertext and invert the transformation.

PIR-based Location Privacy. The PIR concept was originally introduced by Chor et al. [12] and has been extensively studied over years [16, 31–33]. This class of protocols nullifies access pattern attacks since the server doesn't know which item is requested.As discussed in subsection 2.3, there are two broad categories of PIR protocols, *Cryptographic PIR* and *Hardware-based PIR*. Cryptographic PIR categories can be further divided into two groups, which are *information theoretic PIR* and *computational PIR*. The first group is secure even against computationally unbounded adversary with an assumption that there is not collusion among servers. Computational PIR approaches are built based on computational intractability of well-known problems. Even though this class is more practical in the former, it still entails a prohibitive cost of processing.

Secure hardware PIR is the most practical mechanism among all PIR techniques. [10] adopts 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 unstrusted LBS is still able to infer user's location by observing a number of PIR request for each query. This is because that different queries will have different cardinality of private retrievals. As the result, this technique fails to provide strong location privacy. Ghinita et. al. [2] present a 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 requires an excessive processing cost.

Private Advertising System Online and mobile advertising are very promising advertising markets. However, there are many privacy concerns in adopting these advertising models. 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 [34] and Privad [35] 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. The billing procedure (ads view/click report) is performed using cryptographic techniques to prevent other parties to learn which ads are displayed to users. Juels [36] presented another private advertising system focusing on the private distribution of ads. This scheme utilize PIR and mix network to protect users' privacy. Another private advertising system, call RePriv [37], is presented by Microsoft. This system performs personalization by mining browsing behaviour in privacy-conscious manner and provides users with explicit and precise control over the release of private information from their browsers.

Another class of private advertising system focuses on mobile advertising. MoRePriv [15] advocates for OS-level service to solve a conflict of privacy and content personalization on mobile devices. By combining personal preference mining techniques with a privacy filer, MoRePriv can provide necessary information for targeting advertising without compromise user privacy. Nath et. al. presents SmartAds [38], a contextual advertising system on mobile devices. SmartAds guarantees that the only thing the ad server knows is the ad keyword, which is considered as non-sensitive information while all other sensitive information is keep private. Other system, called MobiAd [39], is designed to build users' interest profiles on their phones, download and display relevant ads and reports clicks via DTN protocol. MobiAd maintains users' privacy by keeping user profile on the handset and routing ad reports around many intermediate node to make them anonymous. [13, 14]

introduce flexible framework for personalized advertising in which users can limit the amount of private information they are willing to share with ad server. Despite providing some 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 without revealing users' locations.

6. CONCLUSION

In this paper, we propose a location privacy aware LBA framework PRAD. By utilizing several state-of-the-art techniques in addressing privacy issues such as space-encoding, private information retrieval, homomorphic encryption, our framework can enable the LBAS to serve clients with location-based advertisements without compromising users' location privacy. We intuitively prove the security of our system and evaluate its performance using real-world data set. We claim that our framework is scalable and very practical. As future work, we aim to further develop PRAD to support personalized LBA.

7. REFERENCES

[1] "Worldwide smartphone usage to grow 25% in 2014," Retrieved Sept 2, 2014, from http://www.emarketer.com/Article/Worldwide-Smartphone-Usage-Grow-25-2014/1010920.

[2] 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*, ser. SIGMOD '08, 2008, pp. 121–132.

[3] G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu, "Differentially private spatial decompositions," in *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, ser. ICDE '12, 2012, pp. 20–31.

[4] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, ser. MobiSys '03, 2003, pp. 31–42.

[5] 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*, vol. 19, no. 12, pp. 1719–1733, Dec 2007.

[6] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: Query processing for location services without compromising privacy," in *Proceedings of the 32Nd International Conference on Very Large Data Bases*, ser. VLDB '06, 2006, pp. 763–774.

[7] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," in *Proceedings of the 10th International Conference on Advances in Spatial and Temporal Databases*, ser. SSTD'07, 2007, pp. 239–257.

[8] S. Lee, S. Park, W.-C. Kim, and D. Lee, "An efficient location encoding method for moving objects using hierarchical administrative district and road network," *Inf. Sci.*, vol. 177, no. 3, pp. 832–843, Feb. 2007.

[9] M. L. Yiu, G. Ghinita, C. S. Jensen, and P. Kalnis, "Enabling search services on outsourced private spatial data," *The VLDB Journal*, vol. 19, no. 3, pp. 363–384, Jun. 2010.

[10] A. Khoshgozaran, C. Shahabi, and H. Shirani-Mehr, "Location privacy: Going beyond k-anonymity, cloaking and anonymizers," *Knowl. Inf. Syst.*, vol. 26, no. 3, pp. 435–465, Mar. 2011.

[11] S. Papadopoulos, S. Bakiras, and D. Papadias, "Nearest neighbor search with strong location privacy," in *Proceedings of the 36th International Conference on Very Large Data Bases*, ser. VLDB 2010, 2010.

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

[13] M. Goetz and S. Nath, "Privacy-aware personalization for mobile advertising," Tech. Rep. MSR-TR-2011-92, August 2011.

[14] M. Hardt and S. Nath, "Privacy-aware personalization for mobile advertising," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12, 2012, pp. 662–673.

[15] D. Davidson and B. Livshits, "Morepriv: Mobile os support for application personalization and privacy," Tech. Rep., May 2012.

[16] S. Wang, X. Ding, R. H. Deng, and F. Bao, "Private information retrieval using trusted hardware," in *Proceedings of the 11th European Conference on Research in Computer Security*, ser. ESORICS'06, 2006, pp. 49–64.

[17] S. W. Smith and D. Safford, "Practical server privacy with secure coprocessors," *IBM Syst. J.*, vol. 40, no. 3, pp. 683–695, Mar. 2001.

[18] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," *Journal of the ACM*, vol. 43, pp. 431–473, 1996.

[19] 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*, ser. FOCS '97, 1997, pp. 364–.

[20] D. Asonov and J.-C. Freytag, "Almost optimal private information retrieval," in *Proceedings of the 2Nd International Conference on Privacy Enhancing Technologies*, ser. PET'02, 2003, pp. 209–223.

[21] A. Iliev and S. Smith, "Private information storage with logarithmic-space secure hardware," in *In I-NetSec 04: 3rd Working Conference on Privacy and Anonymity in Networked and Distributed Systems*. IFIP, Kluwer, 2004, pp. 201–216.

[22] T. Wang, K. Lu, L. Lu, S. Chung, and W. Lee, "Jekyll on ios: When benign apps become evil," in *Proceedings of the 22Nd USENIX Conference on Security*, ser. SEC'13. Berkeley, CA, USA: USENIX Association, 2013, pp. 559–572.

[23] D. Hilbert, "Ueber die stetige Abbildung einer Line auf ein Flächenstück," *Mathematische Annalen*, no. 3, pp. 459–460, 1891.

[24] X. Ding, Y. Yang, and R. Deng, "Database access pattern protection without full-shuffles," *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 1, pp. 189–201, March 2011.

[25] Y. Yang, X. Ding, R. H. Deng, and F. Bao, "An efficient pir construction using trusted hardware," in *Proceedings of the 11th International Conference on Information Security*, ser. ISC '08, 2008, pp. 64–79.

[26] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002. [Online]. Available: http://dx.doi.org/10.1142/S0218488502001648

[27] K. Fall, “A delay-tolerant network architecture for challenged internets,” in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM ’03, 2003, pp. 27–34.

[28] T. W. Arnold, C. Buscaglia, F. Chan, V. Condorelli, J. Dayka, W. Santiago-Fernandez, N. Hadzic, M. D. Hocker, M. Jordan, T. E. Morris, and K. Werner, “Ibm 4765 cryptographic coprocessor,” *IBM J. Res. Dev.*, vol. 56, no. 1, pp. 109–121, Jan. 2012. [Online]. Available: http://dx.doi.org/10.1147/JRD.2011.2178736

[29] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, “Secure knn computation on encrypted databases,” in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD ’09, 2009, pp. 139–152.

[30] 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*, vol. 8, no. 6, pp. 863–873, June 2013.

[31] R. Sion, “On the computational practicality of private information retrieval,” in *In Proceedings of the Network and Distributed Systems Security Symposium, 2007. Stony Brook Network Security and Applied Cryptography Lab Tech Report*, 2007.

[32] J. Trostle and A. Parrish, “Efficient computationally private information retrieval from anonymity or trapdoor groups,” in *Proceedings of the 13th International Conference on Information Security*, ser. ISC’10, 2011, pp. 114–128.

[33] F. Olumofin and I. Goldberg, “Revisiting the computational practicality of private information retrieval,” in *Proceedings of the 15th International Conference on Financial Cryptography and Data Security*, ser. FC’11, 2012, pp. 158–172.

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

[35] S. Guha, B. Cheng, and P. Francis, “Privad: Practical privacy in online advertising,” in *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI’11, 2011, pp. 169–182.

[36] A. Juels, “Targeted advertising ... and privacy too,” in *Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer’s Track at RSA*, ser. CT-RSA 2001, 2001, pp. 408–424.

[37] M. Fredrikson and B. Livshits, “Repriv: Re-envisioning in-browser privacy.”

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

[39] H. Haddadi, P. Hui, and I. Brown, “Mobiad: Private and scalable mobile advertising,” in *in The 5th ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch2010*, 2010.