Enabling Privacy-assured Mobile Advertisement Targeting and Dissemination

Zhenkui Shi
City University of Hong Kong
Hong Kong, China
zhenkui.shi@my.cityu.edu.hk

Xiaoning Liu
City University of Hong Kong
Hong Kong, China
xnliu3@cityu.edu.hk

Xingliang Yuan
City University of Hong Kong
Hong Kong, China
xyuancs@gmail.com

ABSTRACT

With the fast growing market of mobile applications, mobile advertising attracts wide attention from both business and research communities in recent years. Targeted mobile advertising aims to analyze user profile and explore user interests so as to deliver ads to potentially interested users and maximize revenue. However, collecting user personal information raises severe privacy concerns. In this paper, we propose a practical targeted mobile advertising service framework while preserving user privacy and enabling accurate targeting. In particular, this framework enables accurate and private user targeting through a privacy-preserving matrix factorization protocol via homomorphic operations. To achieve private ads dissemination, it further adopts the latest advancement of private information retrieval (PIR) to allow the users to obtain accurate ratings and retrieve the most relevant ads without revealing their profiles and accessed encrypted ads. Security and cost analysis are conducted to show that our design achieves strong security guarantees with practical performance.

Keywords

Privacy preserving, mobile targeted advertising, PIR, matrix factorization

1. INTRODUCTION

For the portability, the greatly improved capabilities in computation, connecting, and sensing, mobile phones are becoming increasing popular. Nowadays, the number of global mobile users has exceeded the number of global PC users. The booming market also forces the prosperity of mobile applications (aka apps). As a major revenue, especially for free apps, mobile advertisements attracts many developers to embed advertising in their apps. To maximize the revenue, targeted mobile advertising aims to collect user personal information, analyze user profile and infer user interests to deliver relevant ads. Deployed targeted mobile advertisement systems include iAd [2], AdMob [1], etc.

However, collecting user personal information raises critical privacy concerns. There are some prior studies on how to design systems for private targeted advertising [4, 8, 10]. Yet, most of them face tradeoffs between accuracy and privacy. Several challenges are yet to be explored. The first is how to score and rank relevant ads privately. Scoring relevant ads needs both users' profiles and ads' profiles, and revealing the above information compromises user privacy directly. The second challenge is about ads dissemination. The server or ad network could infer users' preferences through the ads they retrieve. In addition, click report and billing information can also be used to infer users' preferences.

In this paper, we propose a new privacy-preserving targeted advertising architecture based on privacy-preserving matrix factorization and private information retrieval (PIR) techniques. The proposed architecture aims to achieve accurate targeting while protecting users' privacy. To achieve accurate and efficient user targeting, we adopt a widely used mechanism called matrix factorization [16], which supports batch operations to generate user profiles at one time [14]. For privacy, we follow the remarkable progress on privacy-preserving matrix factorization [19, 14]. These new approaches protect both user rating values and rated ads. Each end user just needs to upload a small number of encrypted ratings to the server to form a sparse matrix. The server can then estimate all the users' profiles and ads' profiles, and calculate ratings accurately and privately. Whereas, in most previous works, the ads for ranking are randomly selected and the number of ads are limited due to the communication overhead [8, 4].

Only utilizing the above technique does not lead to a complete targeted mobile advertising service. How to conduct ads dissemination without exposing users' interests is another important question, because the matrix factorization just returns encrypted ratings to the users. To retrieve ads privately, one possible technique would be searchable encryption (SE) [5], which allows users to retrieve encrypted matched ads. However, search and access pattern leaked in SE may reveal statistical information of users' interests, which might be exploited to compromise the users' privacy. Another possible technique is called private streaming search (PSS), which is adopted in a prior private ads targeting system [12]. However, the overhead of PSS queries introduce expensive overhead in mobile phones. Besides, PSS can only support a small sized ads library. To meet requirements in efficiency and privacy, we utilize the latest advancement in PIR [3, 9], which greatly inspire us that PIR can provide

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SCC'17, April 02 2017, Abu Dhabi, United Arab Emirates
© 2017 ACM. ISBN 978-1-4503-4970-3/17/04…$15.00
DOI: http://dx.doi.org/10.1145/3055259.3055269

a practical scheme for private ads dissemination. It protects users' privacy and accessed ads at the ad network side while introducing acceptable communication overhead and computation cost for both the ad network and users. The contributions are summarized as follows:

1. Our design enables accurate targeting with privacy preservation. Users can get accurate ratings of relevant ads which are returned via privacy-preserving matrix factorization, and then privately retrieve most relevant ads via PIR.
2. The proposed ad framework protects both user and ads profiles. Users' profiles are kept locally. The uploaded ratings are calculated locally and encrypted.
3. We present optimization techniques to make our architecture more practical and efficient, i.e., caching and narrowing down the scale of matrix.

The rest of the paper is organized as follows. Section 2 presents the problem statement of targeted mobile advertising. Section 3 reviews preliminaries on secure matrix factorization, PIR, and homomorphic encryption. Section 4 introduces the architecture and adversary model. Section 5 elaborates on how to enable secure and accurate targeting and dissemination. Section 6 describes the related work. Section 7 concludes the whole paper.

2. PROBLEM STATEMENT

2.1 Targeted Mobile Advertising

The current targeted mobile advertising services like AdMob [1] typically contain four major entities, namely, the user, the advertiser, the publisher, and the ad network, as illustrated in Fig. 1.

- **User:** a party who downloads a mobile app and installs it on her mobile device. Afterwards, the user may click on her interested ads.
- **Advertiser:** a party who delivers targeted ads to users across mobile apps, and is willing to pay for this service.
- **Publisher:** a party who owns mobile apps. The publisher is willing to place ads in some assigned screen "real estate", and will be paid for this service.
- **Ad network:** a party who connects advertisers and publishers. The ad network collects ads and their metadata from advertisers and delivers targeted ads to users via registered apps of publishers.

In a certain ad network, the publishers register their apps and embed an ad library within the apps. Hereafter, we refer to the ad library as the client. The client enables the ad network to collect the user personal information, such as device properties, location, demographics, and interested keywords. When an app is launched by the user, the client interacts with the ad network. Based on the collected user behavioral data, the ad network selects relevant ads for that user. When the ads are received, the client selects one or several the most relevant ads to display in the app. When a user clicks on an ad, she will be redirected to the ad loading page served by the advertiser. Meanwhile, the ad network tracks the ad view/click conversation for future targeting and collects the view/click report for billing. At the end of a billing cycle, the ad network charges the advertiser and shares revenue with the publisher. Typically, the advertiser will be charged either for each ad view in the "charge-per-view" model or for each ad click in the "charge-per-click" model [15, 12].

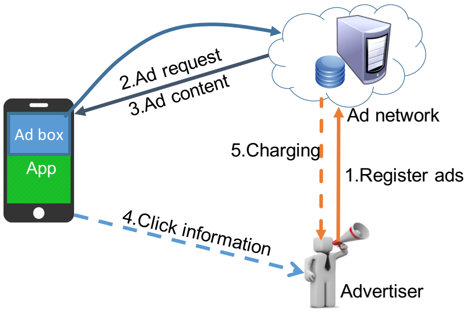

Figure 1: The service flow of targeted mobile ads.

2.2 Design Goals

Privacy. The users' private information should be well protected. Our system aims to enable the ad network to provide most relevant ads without knowledge of the users' source data, behavioral profile, retrieved ads, and ad view history. Here, we do not hide the ad click history from the ad network following the treatment in existing work [22]. Ad clicks are visible to the advertisers, who are highly motivated to share such information with the ad network to improve the quality of ads placement.

Accuracy and Utility. Without loss of user privacy, the accuracy of the retrieved ads should be the same as existing targeted advertising systems. The retrieved ads should match the users with the highest relevant scores. Our system aims to enable the ad network to disseminate the most relevant ads to concerned users. Following existing models, the ad network calculates revenue and generates ad view reports based on the number of ad views and the bid for each ad view [1]. Then it charges the advertisers correctly according to the ad view reports.

Efficiency. The targeted mobile advertising system should be efficient in terms of bandwidth and computation due to the limited mobile battery and expensive cellular network. Besides, the system should minimize the latency of ad fetching which highly affects the user experience.

3. PRELIMINARIES

Privacy-preserving Matrix Factorization., Matrix factorization models map users and items to a joint latent factor space of dimensionality f, such that user-item interactions are modeled as inner products in that space[16, 19, 14]. We adopt the notions as [19]:

- $[n] = \{1, ..., n\}$, $[m] = \{1, ..., m\}$: the sets of users and ads, respectively.
- $\mathbf{u}_i \in \mathbb{R}^d$, $\mathbf{v}_j \in \mathbb{R}^d$: the profile for user i and the profile for ad j, respectively, where d is the dimension of profiles, $1 \leq i \leq n$, $1 \leq j \leq m$.

- $\mathcal{M} \subseteq [n] \times [m]$: the user/ad pairs for which a rating has been generated. $M = |\mathcal{M}|$ is the total number of ratings.
- $r_{ij} \in \mathbb{R}$: the rating generated by user i for ad j.
- $U = \left[u_i^T\right]_{i\in[n]} \in \mathbb{R}^{n\times d}$, $V = \left[v_j^T\right]_{j\in[m]} \in \mathbb{R}^{m\times d}$: the user-profile matrix and the ad-profile matrix, respectively.

Given the ratings in $\mathcal{M}$ which are sparse, the matrix factorization is wished to predict the ratings for user/ad pairs in $[n] \times [m] \setminus \mathcal{M}$ [19]. Given the ratings $\{r_{ij} : (i,j) \in \mathcal{M}\}$, the matrix factorization performs the following regularized least squares minimization:

$$\min_{U,V} \frac{1}{M} \sum_{(i,j)\in\mathcal{M}} (r_{ij} - \langle \mathbf{u}_i, \mathbf{v}_j \rangle)^2 + \lambda \sum_{i\in[n]} \|\mathbf{u}_i\|_2^2 + \mu \sum_{j\in[m]} \|\mathbf{v}_j\|_2^2 \tag{1}$$

for positives $\lambda, \mu \geq 0$, by solving (1), matrix factorization computes user profiles U, ad profiles V, and predicts ratings:

$$r_{ij} = \langle \mathbf{u}_i, \mathbf{v}_j \rangle\,, \quad i \in [n]\,, j \in [m] \tag{2}$$

Gradient descent is used to solve the (1) by iteratively adapting the profiles U and V via the adaptation rule in [16].

Private Information Retrieval (PIR). PIR protocols allow a client to retrieve any object from a database without revealing which object is retrieved [3]. We focus on two typical PIR protocols, namely, computational PIR [3] and information theoretic PIR [9]. In computational PIR protocol, only a computationally bound server is needed. Additively homomorphic public key crypto systems are available to construct such protocols. Information theoretic PIR protocol, in contrast, needs k servers where $k \geq 1$ and these servers should not collude. In this paper, we take the computational PIR as the example to demonstrate our design.

Homomorphic Encryption. Homomorphic encryption enables computation on ciphertexts. Fully homomorphic encryption (FHE) supports arbitrary number of operations [7]. Somewhat homomorphic encryption (SHE) supports a limited number of operations [7]. In this work, we denote fully homomorphic encryption and additive homomorphic encryption by HE and AHE respectively.

4. ARCHITECTURE

4.1 The Architecture Framework

Our proposed framework is illustrate in Fig. 2. There are three entities, namely, client i (a user), the ad network, and the crypto service provider (CSP).

In the client side, there are two components, i.e., a profiling service and an ads service. Inspired by the work about OS support for application personalization and privacy in [6], we introduce the profiling service, in order to leverage the rich local information such as location, text messages, sensors, and browser history as data sources and create an accurate user profile for the ads service and other apps. Meanwhile, the ads service takes the following responsibilities:

• *Calculating and Encrypting.* It uses the profile created by the profiling service and creates a user vector. It calculates the ratings for the predefined ads through the inner product of the user vector and ads profile vectors. Then it encrypts the ratings and uploads them to the ad network.

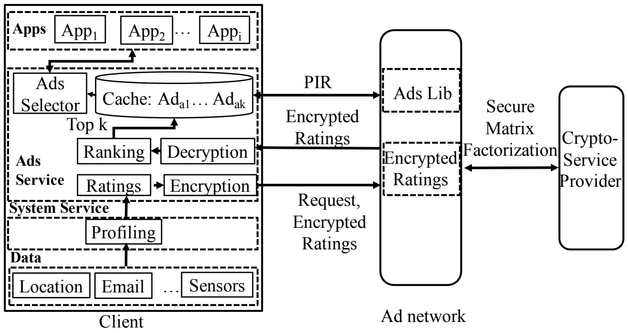

Figure 2: The proposed framework of privacy-assured targeted ads.

• *Decrypting and Ranking.* After receiving the ratings returned by the ad network, it decrypts them and ranks them to get the top k ad indexes with the highest ratings.

• *Prefetching and Retrieving.* It runs the PIR protocol with the ad network to retrieve the most relevant ads.

• *Caching and selecting.* It provides caching service and selects ads for mobile apps.

The ad network communicates with the clients, receives the encrypted ratings, groups the clients and ads, and forms the sparse matrix to be factorized. Then the ad network and CSP do the joint computation for the privacy-preserving matrix factorization. In addition, the ad network also runs the PIR protocol with the clients and sends ads to them. Noted that all the ads are stored in the ad network. The encryption key is stored and used only inside the CSP, which minimizes the risk of key exposure.

4.2 Adversary Model

In this paper, we consider a "honest-but-curious" threat model, which is consistent with existing works on secure advertising [22, 15, 8, 20] and privacy-preserving matrix factorization [19, 14]. The ad network delivers targeted ads to users honestly, yet is curious in learning private information about the users. Specifically, we consider the following information that the users may wish to keep private: source data used for profiling, i.e., behavioral profile, ad view history which is a list of ads that have been displayed to the users. The ad network might collect the users' private information as much as possible either for user targeting, or for malicious purposes like selling user information to third parties. The ad network and CSP honestly follow the protocol we defined. The threats of malicious ads are not considered in this work. Orthogonal mechanisms [21] are proposed to handle these threats.

5. THE PROPOSED DESIGN

In this section, we illustrate our design for targeted mobile advertising, which includes the privacy-preserving matrix factorization protocol and the PIR protocol. And we show how the two protocols cooperate to achieve accurate targeting and ads dissemination.

5.1 Setup

We assume that there are several random selected ads profiles (vectors) which the users' ad services know. Recall that, $\mathcal{M}$ is the formed sparse matrix in the ad network. And we specify the public parameters as follows: (1) We denote the user set and the ad set by $\mathbf{U} = \left\{\mathbf{u}_i | 1 \leq i \leq n, \mathbf{u}_i \in \mathbb{R}^d\right\}$ and $\mathbf{V} = \left\{\mathbf{v}_j | 1 \leq j \leq m, \mathbf{v}_j \in \mathbb{R}^d\right\}$ respectively, where n is the number of users, m is the number of ads, and d is the dimension of profiles. We denote user i and ad j by u_i and v_j respectively. (2) In gradient descent computation, α is the number of bits presenting the fractional part of real numbers in rating, and β is the number of bits presenting the fractional part of real numbers γ. We also leverage the same data structures $\mathbf{U}$, $\mathbf{V}$, $\hat{\mathbf{U}}$, and $\hat{\mathbf{V}}$ defined in [14]. Assume that $\mathcal{B}$ is the message space of HE, the distribution $\mathcal{D}(\mathcal{M}, \mathcal{B})$ works as follows:

For $(i,j) \in \mathcal{M}$, $\mathbf{ev}^{(i,j)} = (ev_1^{(i,j)}, ..., ev_d^{(i,j)})$, where $ev_k^{(i,j)}$ are randomly selected from $\mathcal{B}$. It outputs $\mathbf{ev} = ||_{(i,j)\in\mathcal{M}} \mathbf{ev}^{(i,j)}$. We define the following operations for vectors: $\mathbf{u} \times_c \mathbf{v} = (u_1 v_1, ..., u_k v_k)$, $sum(\mathbf{u}) = \sum_{i=1}^{k} u_i$, and $\langle \mathbf{u}, \mathbf{v} \rangle = sum(\mathbf{u} \times_c \mathbf{v})$, where $\mathbf{u} = (u_1, ..., u_k)$ and $\mathbf{v} = (v_1, ..., v_k)$.

During the setup, u_i generates the public/secret key pairs pk_i and sk_i. The CSP generates two pairs public/secret key pairs for HE and AHE respectively. The ad network specifies the public parameters, i.e., the dimension of profiles d, the bits used to represent the integer λ, the fractional part of a real number in ratings μ, and the fractional part of a real number in gradient descent computation γ.

5.2 Ads Request

In this phase, the clients upload their ratings to the ad network and request to retrieve ad indexes.

1. The client u_i encrypts rating r_{ij} with AHE under the public key of the CSP and then sends $(i, j, \text{AHE}(r_{ij}))$ to the ad network.

2. After receiving the message from the client i, the ad network generates random mask σ_{ij} and then sends $(i, j, \text{AHE}(r_{ij} + \sigma_{ij}))$ to the CSP.

3. Then the CSP decrypts the message, obtains $(i, j, r_{ij} + \sigma_{ij})$, creates the d-dimensional vector $\{(-r_{ij} - \sigma_{ij}, 0, ..., 0)\}$, and sends $\text{HE}(||_{(i,j)\in\mathcal{M}}(-r_{ij} - \sigma_{ij}, 0, ..., 0))$ to the ad network.

4. The ad network builds vectors $\{(\sigma_{ij}, 0, ..., 0)\}$, then obtains $||_{(i,j)\in\mathcal{M}}(-r_{ij}, 0, ..., 0)$ via the following equation:
$$\begin{aligned} &\text{HE}(||_{(i,j)\in\mathcal{M}}(\sigma_{ij}, 0, ..., 0)) + \\ &\text{HE}(||_{(i,j)\in\mathcal{M}}(-r_{ij} - \sigma_{ij}, 0, ..., 0)) \end{aligned} \tag{3}$$

5.3 Privacy-Preserving Matrix Factorization

After users upload encryptions of a small number of ratings to the ad network, the uploaded ratings form a sparse matrix. All the users' ratings in the same matrix are privately calculated via a single privacy-preserving matrix factorization operation. In this phase, the ad network and CSP complete a secure two-party computation protocol [14]. The protocol has the following steps.

1. The ad network performs the following computations:
$$\text{HE}(\mathbf{U}(t-1)) \times \text{HE}(\mathbf{V}(t-1)) - 2^{\alpha} \cdot \text{HE}(\mathbf{r})$$
Then the ad network samples $\epsilon(t-1) \xleftarrow{\$} \mathcal{D}(\mathcal{M}, \mathcal{B})$, adds the mask:
$$\text{HE}(\mathbf{U}(t-1)) \times \text{HE}(\mathbf{V}(t-1)) - 2^{\alpha} \cdot \text{HE}(\mathbf{r}) + \text{HE}(\epsilon(t-1))$$
and sends the ciphertexts to the CSP.

2. The CSP decrypts the ciphertexts then performs computation to get the sum for the inner product $R_{i,j}$, sets $R''(t-1) = ||_{(i,j)\in\mathcal{M}} \ (R_{i,j}, ..., R_{i,j})$, encrypts it and sends $\text{HE}(R''(t-1))$ to the ad network.

3. The ad network computes $\text{HE}(\mathbf{U}'(t))$, $\text{HE}(\mathbf{V}'(t))$, $\text{HE}(\triangledown'_{\mathbf{U}}(t))$ and $\text{HE}(\triangledown'_{\mathbf{V}}(t))$, then samples $\delta_{\mathbf{U}}$, $\delta_{\mathbf{V}}$, $\theta_{\mathbf{U}}$, $\theta_{\mathbf{V}}$ as masks for $\text{HE}(\mathbf{U}'(t))$, $\text{HE}(\mathbf{V}'(t))$, $\text{HE}(\triangledown'_{\mathbf{U}}(t))$, and $\text{HE}(\triangledown'_{\mathbf{V}}(t))$. Then the ad network sends
$$\begin{aligned} &\left\{\text{HE}(\mathbf{U}'(t) + \delta_{\mathbf{U}}(t)), \text{HE}(\mathbf{V}'(t) + \delta_{\mathbf{V}}(t))\right\}, \\ &\left\{\text{HE}(\triangledown'_{\mathbf{U}}(t) + \theta_{\mathbf{U}}(t)), \text{HE}(\triangledown'_{\mathbf{V}}(t) + \theta_{\mathbf{V}}(t))\right\} \end{aligned}$$
to the CSP.

4. The CSP decrypts the ciphertexts, computes
$$\begin{aligned} &\overline{\mathbf{U}'(t) + \delta_{\mathbf{U}}(t)}, \overline{\mathbf{V}'(t) + \delta_{\mathbf{V}}(t)}, \\ &\overline{\triangledown'_{\mathbf{U}}(t) + \theta_{\mathbf{U}}(t)}, \overline{\triangledown'_{\mathbf{V}}(t) + \theta_{\mathbf{V}}(t)} \end{aligned}$$
through fixed point arithmetic, then computes
$$\begin{aligned} &\mathbf{U}''(t) = rec(agg_u \overline{\mathbf{U}'(t) + \delta_{\mathbf{U}}(t)}), \hat{\mathbf{U}}''(t), \\ &\mathbf{V}''(t) = rec(agg_v \overline{\mathbf{V}'(t) + \delta_{\mathbf{V}}(t)}), \hat{\mathbf{V}}''(t), \\ &\triangledown''_{\mathbf{U}}(t) = agg_u \overline{\triangledown''_{\mathbf{U}}(t) + \theta_{\mathbf{U}}(t)}, \\ &\triangledown''_{\mathbf{V}}(t) = agg_v \overline{\triangledown''_{\mathbf{V}}(t) + \theta_{\mathbf{V}}(t)}, \end{aligned}$$
and sends the corresponding homomorphic encryptions $\text{HE}(\mathbf{U}''(t))$, $\text{HE}(\hat{\mathbf{U}}''(t))$, $\text{HE}(\mathbf{V}''(t))$, $\text{HE}(\hat{\mathbf{V}}''(t))$, $\text{HE}(\triangledown''_{\mathbf{U}}(t))$, and $\text{HE}(\triangledown''_{\mathbf{V}}(t))$ to the ad network.

5. The ad network removes mask vectors and gets the homomorphic encryptions $\text{HE}(\mathbf{U}(t))$, $\text{HE}(\hat{\mathbf{U}}(t))$, $\text{HE}(\mathbf{V}(t))$, $\text{HE}(\hat{\mathbf{V}}(t))$, $\text{HE}(\triangledown_{\mathbf{U}}(t))$, and $\text{HE}(\triangledown_{\mathbf{V}}(t))$.

6. The ad network sets the ω_u and ω_v as the thresholds of the user profiles and ad profiles respectively. The corresponding mask vectors are $\mathbf{w}_u$ and $\mathbf{w}_v$. Then, the ad network computes
$$\begin{aligned} &\text{HE}(\triangledown_{\mathbf{U}}(t) \times_c \triangledown_{\mathbf{U}}(t) + \mathbf{w}_u), \text{HE}(\triangledown_{\mathbf{V}}(t) \times_c \triangledown_{\mathbf{V}}(t) + \mathbf{w}_v), \\ &s_u = \omega_u + sum(\mathbf{w}_u), s_v = \omega_v + sum(\mathbf{w}_v) \end{aligned}$$
and sends them to the CSP.

7. The CSP decrypts the ciphertexts, computes
$$\begin{aligned} &s'_u = sum(\triangledown_{\mathbf{U}}(t) \times_c \triangledown_{\mathbf{U}}(t) + \mathbf{w}_u), \\ &s'_v = sum(\triangledown_{\mathbf{V}}(t) \times_c \triangledown_{\mathbf{V}}(t) + \mathbf{w}_v) \end{aligned}$$
and returns the boolean vector $(s_u - s'_u \geq 0?, s_v - s'_v \geq 0?)$ to the ad network. The ad network checks the boolean vector and decides whether to continue the next round matrix factorization computation. If it is true, then the protocol goes to next phase.

5.4 Ranking Ads

In this phase, the user retrieves the ratings of ads, then ranks the ads and obtains the ad indexes which are the most relevant. In this phase, there are the following steps.

1. The ad network samples $m_u, m_v \xleftarrow{\$} \mathcal{D}(\mathcal{M}, \mathcal{B})$, then adds them with homomorphic encryption to get $\text{HE}(\hat{\mathbf{U}}(t) + m_u)$ and $\text{HE}(\hat{\mathbf{V}}(t) + m_v)$. Then sends them to the CSP.

2. The CSP decrypts the ciphertexts and computes

$$\mathrm{HE}(||_{j\in\mathcal{M}_J}(\mathbf{u}_i+\zeta_i), \mathrm{HE}(||_{j\in\mathcal{M}_J}(\mathbf{v}_j+\xi_j)$$

then sends them to the ad network.

3. The ad network removes the masks and obtains $\mathrm{HE}(||_{j\in\mathcal{M}_J}(\mathbf{u}_i)$ and $\mathrm{HE}(||_{j\in\mathcal{M}_J}(\mathbf{v}_j)$. Then the ad network generates random mask vector ψ to compute

$$\mathrm{HE}(||_{j\in\mathcal{M}_J}\mathbf{u}_i \times_c ||_{j\in\mathcal{M}_J}\mathbf{v}_j+\psi)$$

and sends them to the CSP. At the same time, the ad network computes $PE(pk_i, \psi)$ for the user i where pk_i is the public key of the user i.

4. The CSP decrypts the message and computes

$$||_{j\in\mathcal{M}_J}\langle\mathbf{u}_i \times_c \mathbf{v}_j\rangle+\psi',$$

then computes and sends

$$PE(pk_i, ||_{j\in\mathcal{M}_J}\langle\mathbf{u}_i \times_c \mathbf{v}_j\rangle+\psi')$$

to the ad network.

5. The ad network sends $PE(pk_i, \psi)$ and $PE(pk_i, ||_{j\in\mathcal{M}_J}\langle\mathbf{u}_i \times_c \mathbf{v}_j\rangle+\psi')$ to the user i.

6. The user i decrypts the messages, computes ψ' and gets $||_{j\in\mathcal{M}_J}\langle\mathbf{u}_i \times_c \mathbf{v}_j\rangle$.

7. The user i runs the top k algorithm on the ratings $||_{j\in\mathcal{M}_J}\langle\mathbf{u}_i \times_c \mathbf{v}_j\rangle$ to get the k indexes $ai = a1, ..., ak$.

5.5 Private Ads Dissemination

After the user knows the indexes of the most relevant ads, the next task is to retrieve these ads while preventing the ad network from knowing which ads are retrieved and inferring the user preferences. Searchable encryption (SE) enables private data retrieval while its context is that a client tries to outsource its encrypted data to a server [5]. Furthermore, SE may leak information such as index information, search pattern, and access pattern [5]. In the targeted mobile advertising scenario, a malicious ad network can pretend to be a client and the leaked information can be used to infer normal users' preferences. As a generalization of PIR, private streaming search is another available scheme [12]. However, it is more suitable for streaming data retrieval. Inspired by the latest work [9, 3], PIR can be considered as one of the available and practical schemes for private ads retrieval. And it also provides more robust private-preserving scheme compared SE and other schemes.

The computational PIR protocol only needs one single server at the price of heavy computation cost. While information theoretic PIR needs at least two servers and these two servers should not collude. We take the computational PIR as the prototype to illustrate our design for its simple architecture. In our scheme, the ads in the same matrix form a library as a database. The ads library is $L = Ad_1, ..., Ad_n$, where Ad_i is the ad with index i. l is the bit size of Ad_i. l_0 is the bit size that can be used for homomorphic operations. Each ad Ad_i can be split in chunks of l_0 bits $Ad_{i,j}$ as $Ad_i = \{Ad_{i,0}, ..., Ad_{i,l/l_0}\}$ where $j \in [1, ..., (l/l_0)]$. There are three steps in the scheme. The basic workflow of traditional computational PIR is illustrated in Fig. 3. Our design is based on XPIR [3]. It is optimized through aggregation and recursion. The ad network aggregates α ads as a database element. Considering the PIR parameter (n, l, α, k) where k is the recursion parameter, $n' = \lceil n/\alpha \rceil$, and $n'_1 = ... = n'_k = \left\lceil n^{1/k} \right\rceil$, the query is $Q = (Q_1, ..., Q_k)$, and the database elements are $(db_1, ..., db_{n'})$. During query generation, the client generates a query Q_j with the basic PIR protocol to retrieve an element of index i_j in a database of n_j where j in $[1, ..., k]$ and $Q = (Q_1, ..., Q_k)$. During reply, the ad network uses $(db_{(1,i_{j+1},...,i_k)}, ..., db_{(n'_j,i_{j+1},...,i_k)})$ as a database, and computes using the basic PIR protocol for reply. During reply extraction, the client decrypt the k encryption layers to get α elements and return the corresponding element.

Query Construction to retrieve ad Ad_{ak} (the client):

1. For i from 1 to n,
   -if $i \neq ak$, create a random encryption of zero
   -if $i = ak$, create a random encryption of one
2. Send the $q = (pk, q_1, ..., q_{n)}$ to the ad network;

Reply Construction (the ad network): The ad network performs following computations

1. For i from 1 to n,
   - For j from 1 to l/l_0,
   - $R_j = Sum^n_{i=1} q_i * Ad_{i,j}$
2. Return $R = (R_1, ..., R_{l/l_0})$

Ads Extraction (the client):

1. Client decrypts the coordinates of the reply vector R and recover Ad_{ak} as the concatenation of decrypted chunks.

Figure 3: The workflow for privacy-assured ads dissemination.

5.6 Security Analysis

All the private information is strongly protected in each phase of our proposed design. In the phase of ads request, the ratings are calculated locally, and then encrypted by the CSP's public key and sent to the ad network. In the phase of privacy-preserving matrix factorization, the ad network and the CSP execute the secure two-party computation protocol based on homomorphic encryption and random mask techniques. The ad network operates on homomorphic encrypted ciphertext for the gradient descent computation. If it requires computation service from the CSP, the values are masked before sending to the CSP. The ad network and the CSP know nothing about ads' profiles and users' profiles under the adversary model, where they do not collude [14]. In the phase of ads dissemination, XPIR is adopted. The building block is realized on the Ring-LWE based homomorphic encryption scheme [3]. It follows the computational PIR protocol. The homomorphic operation is operated on each ads in the ads database. The ad network knows nothing about the delivered ads.

5.7 Cost Analysis

In the phase of privacy-preserving matrix factorization, we

Table 1: Computation Cost

Entity	Computation Complexity
Ad network	$5dM/L$ (HA + HSM + HM)
CSP	$7dM/L$ (HE + HD)

Table 2: Communication Cost

Entity	Communication Complexity
Client and Ad network	$k \times n^{1/k} \times c + F^k \times l$

mainly focus on the computation cost. Here, the homomorphic operation introduces the major computation cost at the ad network and the CSP side. For a rating matrix formed by m users and n ads, the ad network should take $5dM/L$ homomorphic addition (HA) operations, $5dM/L$ homomorphic scalar multiplication (HSM) operations, and $5dM/L$ homomorphic multiplication (HM) operations. Meanwhile the CSP should take $7dM/L$ homomorphic encryptions (HE) and decryptions (HD) respectively. Here, d is the dimension of the user profile vector, L is the number of slots in the homomorphic scheme based on HELib, and M is the number of ratings namely $n\dot{m}$. The running time is linear in M/L. For example, it will take 192 seconds for the scale of 32786 ratings on a personal computer with 3.4 GHz 6-cores 64GB RAM [14]. That will not be the bottleneck for servers since it is a batching operation for a number of users and ads. In addition, pre-fetching and caching techniques can further improve the user experience.

In the phase of private ads retrieval, XPIR is based on Ring-LWE. And the communication overhead is the major cost for smartphones. This overhead consists of query and result consumption. In our scheme, ads in the same matrix is considered as a database for PIR operation, namely n ads in a database, and the size of each ad is fixed as l bits. With a naive approach, the query overhead is $n \times c$ where c is the size of a ciphertext and the result overhead is about $l \times F$ where F is the expansion factor of the encryption scheme. In practice, one may reduce the query size by aggregating ads and recursion under the parameter k. The corresponding query size and result size are $k \times n^{1/k} \times c$ and $F^k \times l$ respectively [3]. For example, we choose the security parameter for the Ring-LWE encryption as $(1024, 60)$ for the number of coefficients per polynomial and the number of bits per coefficient. And we choose 2.5KB for each ad and $k = 3$. Then the ciphertext for each ad is 16KB and the expansion factor is about 6.4. Given a matrix with 1000 ads, the query size is about 480KB and the result size is about 655.36KB.

The overall cost is summarized as Table 1 and Table 2. We leave the experimental evaluation as the future work.

6. RELATED WORK

Private Targeting and Recommendation. Adnostic proposed by Toubiana et al. [22] is one of the schemes for private targeted advertising. In this scheme, clients retrieve about twenty ads randomly and rank them locally. The accuracy can hardly be guaranteed because of the limited number of returned ads. Kristin et al. [18] proposed to use homomorphic encryption schemes for advertising and pricing. However, this scheme requires a cloud service provider to store all encrypted user profiles uploaded by the clients and all encrypted ads uploaded by the advertiser.

Accurate targeting demands accurate ratings. Privacy-preserving recommendation schemes can be another choice for private rating in mobile targeted advertising. Nikolaenko et. al. [19] proposed the first scheme for privacy-preserving matrix factorization based on garbled circuits and additive homomorphic encryption. And they tried to improve efficiency by multi-thread and parallelization. However, it is not practical because of the excessive communication and computation. Kim et. al. [14] proposed the scheme based on FHE and secure two-party computation. The computation and communication cost is greatly reduced. However, applying privacy-preserving recommendation systems like [19, 14] to our context is not sufficient, because they do not consider ads dissemination.

Private Ads Dissemination. There are several schemes based on anonymization such as Privad and ObliviAd [8, 4]. Privad anonymizes the message flow for ads dissemination by introducing a dealer [8]. ObliviAd uses secure hardware-based PIR and oblivious RAM (ORAM) for private ads dissemination [4].

PIR can be used as a building block for private ads dissemination. However, traditional PIR schemes [11, 17] introduce a large expansion factor that compromises the efficiency [3]. The scheme based on private stream search (PSS) has the similar issue. For example, Jiang et al. [12] adopted PSS for ads dissemination. Although the accessed ads and users' private information are protected, the communication overhead and the size of the ads are large. Later, they investigate privacy-preserving coupon delivery via a secure two-party computation protocol [13]. Yet, the secure two-party computation protocol based on Yao's garbled circuits would introduce heavy computation overhead in our context. In our design, we adopt the latest progress of computational PIR, namely, XPIR for ads dissemination. It introduces smaller communication cost while protecting ads access pattern.

7. CONCLUSION

In this paper, we propose a new design for targeted mobile advertising which aims to protect users' privacy and achieve accurate using targeting. Our designs include ads request, ads ranking, and ads dissemination by leveraging the latest cryptographic progresses on privacy-preserving matrix factorization and PIR respectively. Security and cost analysis are conducted to show our design can achieve strong security guarantees with practical performance.

8. ACKNOWLEDGMENTS

This work was supported in part by Research Grants Council of Hong Kong (Project No. CityU 11276816), the Natural Science Foundation of China under Project 61572412, and Innovation and Technology Commission of Hong Kong under ITF Project ITS/307/15.

9. REFERENCES

[1] *AdMob.* http://www.google.com/admob/.

[2] *iAd.* http://advertising.apple.com/.

[3] C. Aguilar-Melchor, J. Barrier, L. Fousse, and M.-O. Killijian. Xpir: Private information retrieval for everyone. In *Proc. of PETS*, 2015.

[4] M. Backes, A. Kate, M. Maffei, and K. Pecina. Obliviad: Provably secure and practical online behavioral advertising. In *Proc. of IEEE S&P*, 2012.

[5] C. Bösch, P. Hartel, W. Jonker, and A. Peter. A survey of provably secure searchable encryption. *ACM CSUR*, 47(2):18, 2015.

[6] D. Davidson, M. Fredrikson, and B. Livshits. Morepriv: Mobile os support for application personalization and privacy. In *Proc. of ACM ACSAC*, 2014.

[7] C. Gentry. *A fully homomorphic encryption scheme.* PhD thesis, Stanford University, 2009.

[8] S. Guha, B. Cheng, and P. Francis. Privad: Practical privacy in online advertising. In *Proc. of USENIX NSDI*, 2011.

[9] T. Gupta, N. Crooks, W. Mulhern, S. Setty, L. Alvisi, and M. Walfish. Scalable and private media consumption with popcorn. In *Proc. of USENIX NSDI*, 2016.

[10] M. Hardt and S. Nath. Privacy-aware personalization for mobile advertising. In *Proc. of ACM CCS*, 2012.

[11] R. Henry, Y. Huang, and I. Goldberg. One (block) size fits all: Pir and spir with variable-length records via multi-block queries. In *Proc. of NDSS*, 2013.

[12] J. Jiang, X. Gui, Z. Shi, X. Yuan, and C. Wang. Towards secure and practical targeted mobile advertising. In *Proc. of IEEE MSN*, 2015.

[13] J. Jiang, Y. Zheng, X. Yuan, Z. Shi, X. Gui, C. Wang, and J. Yao. Towards secure and accurate targeted mobile coupon delivery. *IEEE ACCESS*, 4:8116–8126, 2016.

[14] S. Kim, J. Kim, D. Koo, Y. Kim, H. Yoon, and J. Shin. Efficient privacy-preserving matrix factorization via fully homomorphic encryption. In *Proc. of ACM ASIACCS*, 2016.

[15] M. S. Kodialam, T. V. Lakshman, and S. Mukherjee. Effective ad targeting with concealed profiles. In *Proc. of IEEE INFOCOM*, 2012.

[16] Y. Koren, R. Bell, C. Volinsky, et al. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[17] T. Mayberry, E.-O. Blass, and A. H. Chan. Efficient private file retrieval by combining oram and pir. In *Proc. of NDSS*, 2014.

[18] M. Naehrig, K. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? In *Proc. of ACM CCSW*, 2011.

[19] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh. Privacy-preserving matrix factorization. In *Proc. of ACM CCS*, 2013.

[20] A. Reznichenko and P. Francis. Private-by-design advertising meets the real world. In *Proc. of ACM CCS*, 2014.

[21] S. Son, D. Kim, and V. Shmatikov. What mobile ads know about mobile users. In *Proc. of NDSS*, 2016.

[22] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas. Adnostic: Privacy preserving targeted advertising. In *Proc. of NDSS*, 2010.