PROTA: A Privacy-Preserving Protocol for Real-Time Targeted Advertising

Yiming Pang*, Bo Wang*, Fan Wu*, Guihai Chen*, and Bo Sheng†
*Shanghai Key Laboratory of Scalable Computing and Systems
Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
†Department of Computer Science, University of Massachusetts Boston
ympang30@gmail.com; wangbo0727@outlook.com; {fwu,gchen}@cs.sjtu.edu.cn; shengbo@cs.umb.edu

***Abstract*—With the widespread use of Internet, online advertising, as a newly emerged way of delivering advertisements, has become the focus of attention. Compared with traditional ways of advertising, real-time targeted online advertising is much more efficient and profitable, taking the advantage of abundant online users' profiles. Advertisements can be delivered to potential users who are actually interested in the ad content, which improves the accuracy of advertising, and thus potentially increases advertisers' profits.**

However, targeted advertising makes use of online users' personal profiles, which raises significant privacy concerns since personal profiles may contain sensitive information. It is interesting but challenging to design a privacy-preserving protocol, which allows advertising platform to effectively deliver ads to interested users, while protecting the users' private information. In this paper, we propose a Privacy-pReserving prOtocol for real-time Targeted Advertising (PROTA). We theoretically prove the privacy properties of PROTA, and show that the system requirements are satisfied. Evaluations are also conducted to demonstrate the feasibility of PROTA.

I. Introduction

As a major economic driver in the Internet economy, advertising has already become an indispensable part of all kinds of websites. With the help of online users' browsing histories, personal profiles and other related information, online targeted advertising can deliver more relevant ads to online users, which increases the click through rate (CTR) of the ads. As a result, ad exchange appears as a platform to match ads and targeted users. RightMedia, DoubleClick and AdECN are such examples.

In a typical advertising profit model, there are mainly three roles involved, namely publisher, advertiser and ad exchange. Publisher is the owner of the web page which the user visits. Advertisers are those who want to display their ads online and are willing to pay for it. Ad exchange brings the above two together and provides a platform for them to negotiate and transact ads. When the user visits a web page with an empty ad slot, the publisher sends an ad request to the ad exchange which it previously registered on. After receiving the ad request, the ad exchange chooses a set of interested advertisers according to the user's profile, which can be formed by browsing histories, cookies and other related information. Then the ad exchange notifies the selected advertisers of the ad slot and ask for a bid. In respond, advertisers send their bids back, which indicate the amount of money they are willing to pay for displaying their ads on the web page. Upon receiving the bids, an auction is conducted by the ad exchange to decide which ad to display based on the bids offered. After that, the ad exchange sends the winning ad back to the publisher. For every ad viewed in the *pay-per-view* (PPV) advertising model or clicked in the *pay-per-click* (PPC) advertising model, the advertiser pays the ad exchange the corresponding price. Meanwhile the ad exchange in turn provides shares to the publisher.

However, the utilization of personal information comes at the cost of privacy leakage since the personal profile of a user may contain some sensitive information that the user is not willing to expose. The key problem is how to design a privacy-preserving protocol which can keep the virtue of targeted advertising, meanwhile guarantee the user's privacy is not compromised. The development of such a protocol faces challenges mainly from two aspects. First, during the matching between the ads and the users, personal profiles of the users should be kept secret due to our privacy request. The intuitive idea is to let the matching process be completed totally on the client side, but it is actually impractical to load the whole database. Second, the privacy request goes throughout the whole process of advertising, which means that related information such as which ad is actually displayed should also be kept secret. This increases the difficulty of bidding and charging for the ad exchange.

In this paper, we mainly focus on designing the privacy-preserving protocol PROTA to comprehensively protect the user's privacy, while keeping the desiring properties of real-time targeted advertising. PROTA utilizes Bloom filter to enable the matching between the ad and the user to be

This work was supported in part by the State Key Development Program for Basic Research of China (973 project 2014CB340303), in part by China NSF grant 61422208, 61472252, 61272443 and 61133006, in part by Shanghai Science and Technology fund 15220721300, in part by CCF-Tencent Open Fund, in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, and in part by Jiangsu Future Network Research Project No. BY2013095-1-10. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

F. Wu is the corresponding author.

978-1-4673-8590-9/15/$31.00 ©2015 IEEE

completed totally on the client side, meanwhile uses homomorphic cryptograhy to make sure that the sensitive information remains private later in the process. The essence of PROTA lies in our delicate design which can protect the user's privacy in each step of targeted advertising.

Our contributions are summarized as follows:

- To the best of our knowledge, PROTA is the first comprehensive privacy-preserving protocol for real-time targeted advertising which can protect users' privacy thoroughly without sacrificing the accuracy of advertising.
- We propose a well-designed ad delivery mechanism to ensure the ad security in the delivering process, and the charging correctness is guaranteed by the design of e-coin which further enhances our privacy model.
- We implement PROTA and extensively evaluate its performance, which perfectly demonstrates the feasibility of PROTA.

The remainder of this paper is organized as follows. In Section II, some necessary preliminaries are provided. In Section III, we present the detailed system design of PROTA, followed by performance evaluation and related analysis in Section IV. In Section V, we introduce some related works of privacy-preserving mechanism design. Finally, the work is concluded in Section VI.

II. Preliminaries

In this section, we first present the system model for our design. Then we introduce two technical tools that will be used in our design, namely Bloom filter [1] and Boneh-Goh-Nissim cryptosystem (BGN) [2].

A. System Model

In PROTA, we model the procedure of advertising as a typical targeted advertising pattern, which mainly includes the following parts:

Client: Clients refer to all the online users visiting some web pages. Each client c maintains a keywords profile of their own, denoted as $\mathcal{KW}_c = \{kw_1, \cdots, kw_{n_c}\}$, which can be obtained by analyzing the client's demographics, browsing history, information stored in cookies and so on.

Publisher: Publishers own some web pages and are willing to sell the ad slots. The role of the publisher is not vital in our design and it only appears at the requesting and charging stages, which have nothing to do with privacy issues.

Advertiser: Advertisers are those who want to display online ads and are willing to pay for them. Similar to the clients, each advertiser a maintains a keywords profiles according to features of the ads they possess, denoted as $\mathcal{KW}_a = \{kw_1, \cdots, kw_{n_a}\}$. Advertisers can upload their keywords profile to the ad exchange and update them periodically.

Ad Exchange: Ad exchange brings together sellers and buyers of ad slots, *i.e.*, publishers and advertisers. It collects ads (and payments) from advertisers and places them on publishers' web pages (along with paying the publishers).

Trusted Third Party (TTP): A trusted third party (TTP) is introduced in our protocol. The TTP cooperates with the ad exchange to finish the whole process of targeted advertising and supervises potential malicious actions. To make sure that the TTP is secure and does not collude with other parties, it should be ran by supervision organizations such as privacy and consumer advocacy groups *e.g.*, Electronic Privacy Information Center. Moreover, since the TTP is not the beneficiary in the pre-defined advertising profit model, ad exchange should also provide shares to the TTP, which motivates the supervision.

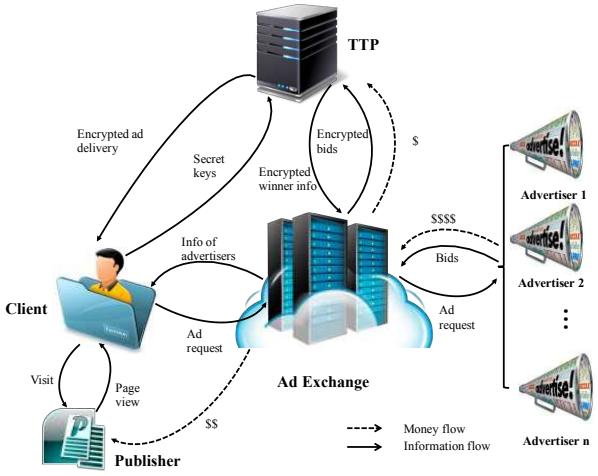


Fig. 1. Typical Model for Online Advertising

For the charging method, we adopt the *pay-per-click* (PPC) model, which means for each ad clicked, the corresponding advertiser will pay.

Fig. 1 illustrates the system framework, including the information flow (denoted by solid arrows) and money flow (denoted as dashed arrows).

B. Bloom Filter

A Bloom filter is a space-efficient probabilistic data structure, first conceived by Burton Howard Bloom in 1970, which provides a way to probabilistically encode set membership using a small amount of space, even when the universe set is huge. False positive matches are possible but false negatives never happen, thus a Bloom filter can achieve 100% recall rate. By setting relevant parameters carefully, the false positive rate can be very low which can be totally neglected.

The definition of Bloom filter is as follows [3]:

Definition 1. *A (k, m)-Bloom filter is a collection of hash functions $\{h_i\}_{i=1}^{k}$, with $h_i : \{0,1\}^* \to [m]$ for all i, together with an m-bit array $B = \{b_j\}_{j=1}^{m}$. If $a \in \{0,1\}^*$, then to insert the element a into this structure, for all $i \in [k]$, $b_{h_i(a)}$ is set to be 1. Then to determine whether $a \in S$ or not, one examines the value of $b_{h_i(a)}$ for each $i \in [k]$ and returns true if all have the value of 1, or returns false if any of $b_{h_i(a)}$ is 0.*

Bloom filter has a strong space advantage over other data structures for representing sets at the cost of risking false positives. Therefore, we need to analyze the total size of a (k, m)-Bloom filter to estimate the parameter setting. In our

situation, the hash functions h_i will be modeled as uniform, independent randomness.

Theorem 1. *Let $(\{h_i\}_{i=1}^{k}, \{b_j\}_{j=1}^{m})$ be a (k, m)-Bloom filter as described in Definition 1. Suppose the filter has been initialized to store some set S of size n. Assume also that $m = \lceil cnk \rceil$ where $c > 1$ is a constant. Then for any $a \in \{0,1\}^*$, the following statement holds true with probability $1 - neg(k)$, where the probability is over the uniform randomness used to model the h_i:*

$$(a \in S) \Leftrightarrow (b_{h_i(a)} = 1, \forall i \in [k])$$

It can be easily proved by a series of independent Bernoulli trials and the detailed proof is neglected. Guaranteed by the theorem, Bloom filter can achieve high performance with extremely low overheads and allowable errors.

C. Homomorphic Cryptosystem

Homomorphic cryptosystem is a special type of cryptosystems, which enables specific types of computations to be conducted on ciphertexts and obtains a new ciphertext, which can be decrypted to match the result of computation applied directly on the original plaintext.

In our study, we adopt BGN cryptosystem [2], which belongs to partially homomorphic cryptosystems. The BGN cryptosystem utilizes a bilinear pairing to allow the computation of a single homomorphic multiplication of two ciphertexts, while still retaining the additively homomorphic properties of earlier cryptosystems.

The definition of bilinear pairing is as follows:

Definition 2. *Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of order n with g, a generator of $\mathbb{G}$. A map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is said to be bilinear if $e(g, g)$ is a generator of $\mathbb{G}_1$ and*

$$e(u^a, v^b) = e(u, v)^{ab} \tag{1}$$

for all $u, v \in \mathbb{G}$ and all $a, b \in \mathbb{Z}$.

Based on bilinear pairing, we can present the BGN cryptosystem as follows [4]:

Definition 3. *Boneh-Goh-Nissim (BGN) Cryptosystem*

Randomly choose two distinct odd primes p and q and let $n = pq$, then let $\mathbb{G}$ with a random generator of g, $\mathbb{G}_1$ be two multiplicative groups of order n with a bilinear pairing $e : (\mathbb{G} \times \mathbb{G}) \to \mathbb{G}_1$. Suppose h is a random generator of the subgroup of $\mathbb{G}$ of order p, then let $T < q$, then $\mathcal{P} = \mathbb{Z}_T$, $\mathcal{C} = \mathbb{G}$, $\mathcal{R} = \mathbb{Z}_n$, and $\mathcal{K} = \{(n, p, q, T, \mathbb{G}, \mathbb{G}_1, e, g, h)\}$ where $(n, p, q, T, \mathbb{G}, \mathbb{G}_1, e, g, h)$ are defined as above.

- *Gen: Given the security parameter ϵ, $Gen(\epsilon)$ generates two distinct $\frac{\epsilon}{2}$-bit primes p, q, sets $n = pq$ and selects a positive integer $T < q$. $Gen(\epsilon)$ then generates two multiplicative groups $\mathbb{G}$, $\mathbb{G}_1$ of order n, that support a bilinear pairing $e : (\mathbb{G} \times \mathbb{G}) \to \mathbb{G}_1$, as well as random generators $g, u \in \mathbb{G}$, and sets $h = u^q$ such that h is a generator of the subgroup of order p. The public key is $(n, g, h, \mathbb{G}, \mathbb{G}_1, e)$, and the private key is p.*
- *Enc: Given a message $m \in \mathcal{P}$ and a public key pk, $Enc(pk, m)$ chooses a random $r \in \mathcal{R}$ and the ciphertext can be calculated as*

$$c = g^m h^r \ mod \ n \tag{2}$$

- *Dec: Given a ciphertext $c \in \mathcal{C}$ and a private key sk, $Dec(sk, c)$ calculates the plaintext as*

$$c' = c^p = (g^p)^m \ mod \ n \tag{3}$$

and uses Pollard's lambda method [5] to take the discrete logarithm of c' in base g^p

There are some homomorphic properties of BGN cryptosystem. Let $c_1 = g^{m_1} h^{r_1} \mod n$ and $c_2 = g^{m_2} h^{r_2} \mod n$. Then

$$c_1 c_2 \mod n = g^{m_1} h^{r_1} g^{m_2} h^{r_2} = g^{m_1+m_2} h^{r_1+r_2} \mod n \tag{4}$$

is a valid encryption of $m_1 + m_2$, which demonstrates the additive homomorphism of the BGN cryptosystem. Similarly, substraction of encrypted messages and constants can be accomplished by computing $c_1 c_2^{-1} \mod n$.

In addition to additive homomorphic operations, the BGN cryptosystem also allows a single homomorphic multiplication of plaintexts. With the bilinear pairing e, set $g_1 = e(g, g)$ and $h_1 = e(g, h)$, since g generates $\mathbb{G}$, it holds that $h = g^\alpha$ for some α. Given c_1, c_2, and a random $r \in \mathcal{R}$, a ciphertext representing the product $m_1 m_2$ can be calculated by

$$\begin{aligned} e(c_1, c_2) h_1^r &= e(g^{m_1} h^{r_1}, g^{m_2} h^{r_2}) h_1^r \\ &= e(g^{m_1} g^{\alpha r_1}, g^{m_2} g^{\alpha r_2}) h_1^r \\ &= e(g^{m_1+\alpha r_1}, g^{m_2+\alpha r_2}) h_1^r \\ &= e(g, g)^{(m_1+\alpha r_1)(m_2+\alpha r_2)} h_1^r \\ &= g_1^{m_1 m_2 + m_1 \alpha r_2 + m_2 \alpha r_1 + \alpha^2 r_1 r_2} h_1^r \\ &= g_1^{m_1 m_2} h_1^{m_1 r_2 + m_2 r_1 + \alpha r_1 r_2 + r} \\ &= g_1^{m_1 m_2} h_1^{\overline{r}} \end{aligned} \tag{5}$$

where $\overline{r}$ is a uniformly random element of $\mathcal{R}$. By replacing g and h with g_1 and h_1 respectively, further additions are still possible on the resulting ciphertext. Further multiplications are not possible as there is no pairing defined from $\mathbb{G}_1$ to another isomorphic group.

The ability to perform simple deterministic computations on encrypted data makes homomorphic cryptosystems ideal for creating privacy-preserving protocol.

III. System Design

In this section, we thoroughly introduce the design of our privacy-preserving protocol PROTA. We first provide an overview of our system with the design rationale. Then the detailed design is presented step by step, followed by some analysis.

A. Design Rationales

Upon the system level, we specify some system goals which should be satisfied to make the system practical [6].

Performance: Performance always comes first and we must guarantee the computation and communication overheads of the system are acceptable.

The bottleneck of the system performance is at the client side since the computational power of other parts is based on large-scale web servers. Therefore, we adopt Bloom Filter to accelerate the matching stage which much improves the system performance.

Effectiveness of Data Acquisition: Since privacy-preserving protocol inevitably hides some useful information from the ad exchange, we should ensure that some data such as click-through rate (CTR) of some specific ads should be possible to calculate under the protocol setting without compromising the clients' privacy.

In our design, the final results of all clicked ads are returned to the ad exchange in plaintext, but all the results are anonymous since there is no client ID attached.

Click-Fraud Detection: Generally speaking, click-fraud detection is inherently in conflict with privacy since privacy-preserving protocol hides the client and preserves them from being tracked.

In PROTA, the TTP can monitor the behavior of all clients by checking the recieving e-coins, which can potentially help click-fraud detection.

More importantly, we also formally define the privacy goals of privacy-preserving targeted advertising [7].

Profile Privacy: No one can obtain exact information of the client's profile and no one in the system can associate any unit of learned information (*e.g.*, clicked ads) with any client's personally identifying information (PII).

Profile Unlinkability: Profile unlinkability refers to the property that the adversary cannot associate separate units of learned information with a single client.

Then we define security properties for the client and the advertiser in the system.

Client: The keywords profile of the client is the pivotal issue in the protocol. Besides the privacy of the keywords profile, we are also supposed to guarantee that which ad the client clicked on should be kept secret.

Advertiser: In [8], advertisers' bids are also considered as privacy since an advertiser's bid can reflect his marketing strategy. In our protocol, the security of advertisers' bids is not our primary task. However, the advertisers' bids in our protocol are secure and won't be learned by other parties due to the properties of homomorphic cryptosystem.

B. Detailed Designs

Before invoking PROTA, some initialization works need to be accomplished for each party.

Client: As we described before, each client c maintains a keywords profile, denoted as $\mathcal{KW}_c = \{kw_1, \cdots, kw_{n_c}\}$. The keywords profile of the client can be updated periodically according to the client's online history. The update does not influence PROTA. Besides, before each round of advertising, the client generates a symmetric key key_c for ad delivering phase and keeps it secret.

Advertiser: Each advertiser is assigned a unique ID denoted as Ad_i. Similar to the client, each advertiser Ad_i also maintains a keywords profile, denoted as $\mathcal{KW}_{Ad_i} = \{kw_1, \cdots, kw_{n_{Ad_i}}\}$, which also can be updated periodically.

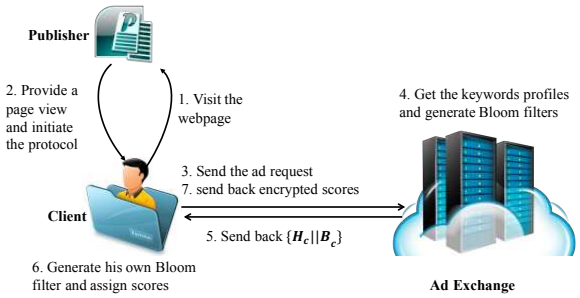


Fig. 2. Matching Phase

Ad Exchange: The ad exchange holds the keywords profile of all registered advertisers, denoted as $KW = \{\mathcal{KW}_{Ad_1}, \cdots, \mathcal{KW}_{Ad_n}\}$. According to the BGN cryptosystem, the ad exchange generates its public key pk_{AE} and secret key sk_{AE}.

TTP: Similar to the ad exchange, the TTP publishes its public key pk_{TTP} and keeps its secret key sk_{TTP} private. What's more, a secret symmetric key key_t is also required for the TTP.

The size of the universal keywords set is denoted as N_{kw}. The information each party holds before the protocol is evoked is summarized in Table I

TABLE I
INFORMATION DISTRIBUTION

	Information Possessed	
System Part	Public	Private
Client c	pk_{TTP}, pk_{AE}	$\mathcal{KW}_c, key_c$
Ad Exchange	pk_{TTP}, pk_{AE}, KW	sk_{AE}
Advertiser Ad_i	$pk_{TTP}, pk_{AE}, \mathcal{KW}_{Ad_i}$	Bidding Strategy
TTP	pk_{TTP}, pk_{AE}	sk_{TTP}, key_t

Now, we are ready to present the detailed designs of PROTA in four phases: matching, bidding, delivering and charging, which will be introduced one by one.

1) Matching Phase: When a client c visits a publisher's website with an ad slot, the publisher provides the client with a page view and initiates the protocol (Fig. 2). Then the client sends an ad request $Ad\_request_c$ to the ad exchange. Upon receiving $Ad\_request_c$, the ad exchange gathers the keywords profile of current active advertisers, the amount of which is represented as n_{act}. The universal set of all the keywords profiles of current active advertisers is represented as $KW_{act} = \{\mathcal{KW}_{Ad_i} | Ad_i \text{ is active}\}$.

Based on the definition of Bloom filter which is described in Section II-B, the ad exchange generates k hash functions $H_c = \{h_i\}_{i=1}^{k}$ for client c. After obtaining the hash function set, for each active advertiser, the ad exchange generates a (k, m)-Bloom filter, where $m = \lceil cN_{kw}k \rceil, c > 1$ is a constant. Now we have a set of Bloom filters for client c, denoted as $B_c = \{B_{Ad_1}, \cdots, B_{Ad_{n_{act}}}\}$, where $B_{Ad_i} = \{b_j\}_{j=1}^{m}$ is the corresponding Bloom filter for the keywords profile of advertiser Ad_i. Then the ad exchange sends $\{H_c||B_c\}$ back to the client c to allow the client to match the advertisers.

Once receiving $\{H_c||B_c\}$, the client c has the ability to match the advertisers locally. First, the client generates the Bloom filter $\mathcal{B}_c$ of his own keywords profile $\mathcal{KW}_c$ according to the hash function set H_c. Then for advertiser Ad_i, the client determines for each $kw_j \in \mathcal{KW}_c$ whether $kw_j \in \mathcal{KW}_{Ad_i}$ by testing if the following statement stands:

$$b_{h_i(kw_j)} = 1 \text{ for } \forall i \in [k]$$

Suppose that there are in total f_{c,Ad_i} (out of n_c) keywords of client c found in advertiser Ad_i's keywords profile, we can calculate a score for the advertiser by the following method:

$$sc_{c,Ad_i} = \frac{f_{c,Ad_i}}{n_c} \tag{6}$$

sc_{c,Ad_i} reflects the matching degree between client c and advertiser Ad_i which ranges from 0 to 1. However, since BGN encryption must be applied to integers, here we must transform the scores into integers. We do the following alternation:

$$sc_{c,Ad_i} = [\frac{f_{c,Ad_i}}{n_c} \times 1000] \tag{7}$$

By this method, we keep three decimal places, which preserves the level of precision appropriately. Therefore, sc_{c,Ad_i} indicates the potential click probability of client c on advertiser Ad_i. Under normal conditions, clients are inclined not to expose which advertisers they are interested in, which makes the matching degree also a part of privacy. Therefore, instead of directly sending the scores to the ad exchange, the client sends the score information $\mathcal{SC}_c = \{sc_{c,1}, \cdots, sc_{c,n_{act}}\}$ encrypted by the public key of the TTP to the ad exchange as folllows:

$$(\mathcal{SC}_c)_{pk_{TTP}} = (sc_{c,1}, \cdots, sc_{c,n_{act}})_{pk_{TTP}}$$

In this way, both the keywords profile and the matching information of client c will not be leaked to any party during the matching phase.

2) Bidding Phase: In fact, the bidding phase and the matching phase can be conducted simultaneously. Once the ad exchange sends $\{H_c||B_c\}$ to client c, it can start the bidding phase by sending each active advertiser a bid request $Bid\_request_{Ad_i}$, which includes the information of the publisher since advertisers can bid more reasonably according to the publisher information.

In respond, each advertiser sends the bid back to the ad exchange. Similar to the client, advertisers also send encrypted bid by the public key of the TTP, denoted as $(bid_{c,Ad_i})_{pk_{TTP}}$.

So far, the ad exchange has gathered the "matching score" and "paying bid" for each advertiser. These two together form the "ultimate advantage" of each advertiser. Notice that the "matching score" defined in Section III-B1 is already normalized to the range of $[0, 1000]$, we can just multiply them together to get the "final bid" as shown in Eq. 8. The multiplication perfectly makes sense since if an advertiser has no keywords matching the client, the final bid of the advertiser will be set to zero which makes him out of the auction automatically.

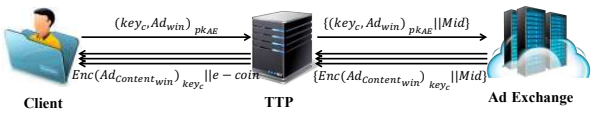


Fig. 3. Delivering Phase

$$\begin{aligned}&(sc_{c,Ad_i})_{pk_{TTP}} \times (bid_{c,Ad_i})_{pk_{TTP}} \\ &= (sc_{c,Ad_i} \times bid_{c,Ad_i})_{pk_{TTP}} = (fbid_{c,Ad_i})_{pk_{TTP}}\end{aligned} \tag{8}$$

Then the ad exchange sends the final bids $\{fbid_{c,Ad_1}, \cdots, fbid_{c,Ad_{n_{act}}}\}$ to the TTP.

Now TTP can decrypt the ciphertext and get the final bid by sk_{TTP}. The TTP runs a second price auction and the one with the highest $fbid$ wins the auction. To ensure the truthfulness of the auction, we need to make sure that the winner pays less than his bid. To solve this problem, we can let the winner pay $fbid_{c,second}/1000$, which is the final bid of the second highest bidder. The truthfulness is guaranteed by:

$$\frac{fbid_{c,second}}{1000} < \frac{fbid_{c,win}}{1000} \leq bid_{c,win}$$

So far the TTP already knows the winner of this round, however, the TTP still has no information of the ad itself since the only information it possesses is the ID of the winner, which still preserves the privacy of the client c.

3) Delivering Phase: In the delivering phase, we use a method similar to that used in [9] during the ad dissemination, which keeps the ad delivered to the client secret both from the TTP and the ad exchange. The delivering protocol is shown in Fig. 3.

First of all, the TTP notifies the client c of the winning advertiser's ID Ad_{win}. Recall that during the initialization, each client generates a symmetric key key_c, the client c encrypts the symmetric key along with the winning advertiser's ID by the public key of ad exchange to get the ciphertext as $(key_c, Ad_{win})_{pk_{AE}}$, then sends this message to the TTP. Upon receiving the message, the TTP assigns the message a message ID Mid and stores a mapping between Mid and client ID c. Then the TTP appends the Mid to the message and forwards it to the ad exchange, as $\{(key_c, Ad_{win})_{pk_{AE}}||Mid\}$.

By the secret key sk_{AE}, the ad exchange can decrypt the message and find the ad corresponding to the ID Ad_{win}. Then by the symmetric key key_c sent by the client c, the ad exchange encrypts the ad content with it and sends it back to the TTP along with the message ID Mid, which the TTP uses to lookup the client to forward the ad to.

Now, the symmetric key possessed by the TTP is on the stage. Since we require secret charging to protect the client's privacy, we design a " e-coin" protocol as in [6] to help ensure the correctness of the charging phase and preserve the privacy. The e-coin is generated by the TTP as

$$e - coin_{win} = sig(Enc(Ad_{win}, charge_{win})_{key_t}, t)_{sk_{TTP}}$$

It consists of a digital signature produced by the TTP on (i) the ciphertext obtained by encrypting the winner's ID Ad_{win}

and the corresponding price calculated as $fbid_{second}/1000$, with the symmetric key key_t chosen by the TTP, and on (ii) a timestamp t, which ensures the validity of the e-coin.

After generating the e-coin of the winning advertiser, the TTP forwards the encrypted ad content sent from the ad exchange along with the e-coin of the winner to the client c. Finally, the client gets the encrypted ad content which can be decrypted and displayed.

4) Charging Phase: According to the design goals of our system, we aim at keeping the ad clicked by the client secret to the ad exchange. However, the goal is in conflict with the traditional charging method. Therefore, we design a "periodic" charging method with the help of the e-coin we introduced before.

In PROTA, once the client clicks on the ad displayed, the client automatically sends the associated e-coin to the ad exchange. The e-coin is actually the "receipt voucher" of the ad exchange. Different from the normal way of collecting money, the ad exchange has a billing period, which means that it can only charge the advertisers at the end of each billing period. When a billing period is over, the ad exchange sends all of the e-coins gathered during this billing period to the TTP. The TTP first verifies the validity of these e-coins and drops those illegal ones, then decrypts them by the secret symmetric key key_t to get the ad IDs and corresponding charges. Then for each advertiser, the TTP calculates the amount of money it has to pay for this period and sends the message to the ad exchange. Finally, the ad exchange can collect money from advertisers and provide shares to the publisher and the TTP.

C. Security Analysis

In Section III-A, we define the privacy goals of PROTA: *Profile Privacy* and *Profile Unlinkability*. We will discuss them respectively.

1) Profile Privacy: In terms of profile privacy, during the matching phase, the matching process is completed totally on the client side, the way of which will not leak any information about the client to other parties in the system. Then the client sends the matching score to the ad exchange in encrypted form by the public key of the TTP, which means that the ad exchange cannot learn any information about the matching results of the client.

During the bidding phase, all of matching scores, bids and final bids are in the ciphertext space, which makes it impossible for the ad exchange to extranct any information from the messages. The TTP can decrypt the final bids and get the winner by running an auction but the advertiser ID itself reveals nothing about the ad, which gurantees that the client's privacy is not compromised.

In the delivering phase, the delivery protocol can separate the ad content and the client ID at the TTP. Hence, the TTP knows the ID of the winning advertiser but cannot associate the ad contents to the client's PII, which in this case is the client ID, satisfying the requirements of profile privacy. As for the ad exchange, the only information it can get is someone has requested for an ad.

For the charging phase, we adopt the e-coin method with periodically charging. Due to the charging latency, which ad is actually clicked by a particular client is unknown to the ad exchange. It is interesting to observe that the degree of privacy is determined by the number of e-coins that are provided by non-compromised clients in the respective mixing procedure. Notice that a malicious ad exchange could in principle derive a particular client's profile by allowing only the e-coins of that honest client to reach the TTP. The resulting bill would reveal the client's profile, thus breaking the desired privacy property. Typical ad exchanges, however, behave rationally, *i.e.*, their primary goal is to excel in commerce rather than to identify clients at all cost. Therefore, PROTA can protect the privacy of clients' profiles against rationally-behaving ad exchange.

In conclusion, we can guarantee that profile privacy is satisfied in the whole procedure.

2) Profile Unlinkability: In our design of PROTA, not only is the ad exchange not able to learn which client clicks on which ad, but it also cannot learn whether or not two or more ads were seen by the same client. This property is enforced by *(i)* the structure of the e-coins, which is unlinkable and does not reveal any information about the client's identity or profile, and *(ii)* the mixing, which breaks the correlation between the billed ads and received e-coins. Breaking this correlation is crucial since the attacker may learn the correlation between e-coins and clients by looking at the traffic on the non-anonymous communication channel between clients and the TTP.

In conclusion, all the design goals are met in our design.

IV. Evaluation

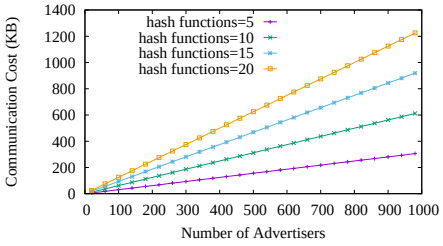
In this section, we implement our design and evaluate the performance of our system PROTA.

A. Methodology

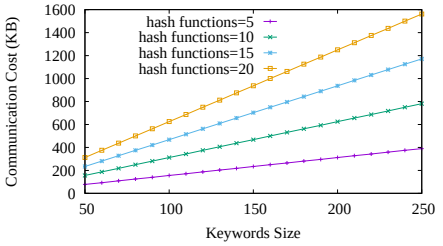
We simulate the system of PROTA on a laptop with Intel® Core™ i5-2410M 2.3GHz GPU. The operating system environment of the experiment is Linux Ubuntu 12.04. The overall configuration is in line with most client's devices, which means that the evaluation results can represent the average well. In our design, we fully implement the BGN cryptosystem with Paring-Based Cryptography library (PBC library) and we simulate the computations and communications on the client, the ad exchange and the TTP. Usually, the computational power of the ad exchange and the TTP is based on large-scale web servers, which means their computational power can be regarded as ideal. Therefore, in our evaluation part, we mainly concentrate on the client side, whose computational power is often limited, and demonstrating the feasibility of our system on the client side.

B. Simulation Results

During the matching phase, the ad exchange sends all the Bloom filters of interested advertisers and the set of hash functions to the client, which generates relatively large communication costs. We simulate the whole matching phase and measure the overall communication costs. The constant



(a) Communication cost of different number of advertisers



(b) Communication cost of different size of keywords

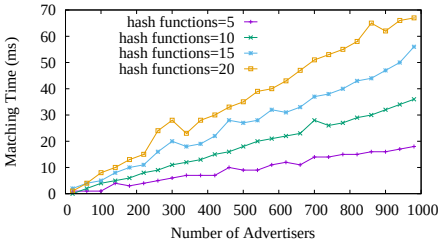
Fig. 4. Communication cost in matching phase

parameter c in Bloom filter is set to be 5 in all cases, which is a reasonable presumption. We separately measure the communication costs of the matching phase when the number of hash functions is set to be 5, 10, 15 and 20. Under this setting, we can guarantee that the false positive rate is lower than 0.5% in the worst scenario (where c is set to be 5 and k is set to be 5). Fig. 4 shows the communication costs of different number of advertisers and different size of keywords during the matching phase.

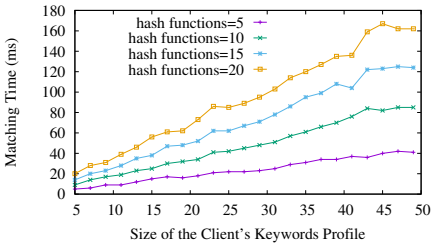
For Fig. 4(a), we set the size of universal keywords set to be 100, which is a reasonable assumption since the top-level categories on Google Ads are only 27. We suppose that 100 keywords can basically cover all the usual ad categories. According to the figure, we can see that the communication costs are strictly linear to the number of advertisers. Even in the most extreme case, where k is set to be 10 and the number of advertisers is 1000, the communication cost does not exceed 1.4MB, which can be handled by most networks. However, setting k to be too large sometimes is useless, since it only reduces the false positive rate a little with generating relatively high communication and computation overheads to the client. In average, the communication cost is about 200KB-600KB.

Similarly, in Fig. 4(b), we fix the number of advertisers to be 500 and alter the size of the universal keywords set. It is obvious that the communication cost is still within the acceptable range even in the most extreme case.

To finish the matching, the computation on the client side includes running the hash functions to match the advertisers and encrypting the scores with BGN cryptosystem. To illustrate the difference of computation time in different parts, we separately measure the computation time of the matching part and the encrypting part. Fig. 5 shows the computation time of the matching part. In Fig. 5(a), the size of the client's keywords profile is set to be 10 and in Fig. 5(b), the number of advertisers is set to be 500. For both cases, the size of the

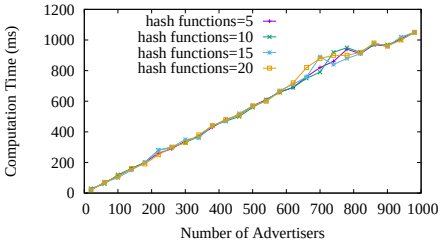


(a) Matching time of different number of advertisers

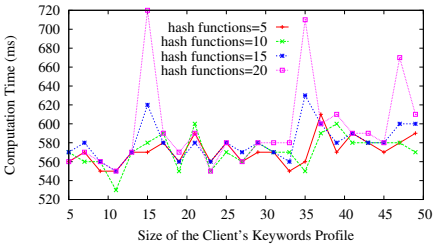


(b) Matching time of different size of profile

Fig. 5. Matching time in matching phase



(a) Computation time of different number of advertisers



(b) Computation time of different size of profile

Fig. 6. Computation time in matching phase

universal keywords profile is 100.

Clearly, in all cases the matching time does not exceed 200ms which means the matching process doest not take long time and doest not hamper the user experience.

Then we measure the total computation time during the matching phase, as shown in Fig. 6.

We use the same setting as measuring the matching time. In Fig. 6(a), we can observe that the total computation time during the matching phase almost has nothing to do with the choice of k, which means the encryption time dominates the whole computation process. In Fig. 6(b), we can also see that the total computation time is nearly irrelevant to the choice of k and the size of the client's keywords profile. The

inordinance is the result of the randomization introduced in the cryptographic algorithm.

In conclusion, we implemente BGN cryptosystem based on PBC library and simulate the process of PROTA. We evaluate the computation time and communication costs on the client side, which proves the feasibility of PROTA.

V. Related Works

Privacy issue in targeted advertising has always been a hot topic since the day targeted advertising was born. Juels [10] was the first one to explore the concept of privacy-preserving targeted advertising and to propose several technical solutions. Juels designed a full *mix networks* mainly based on private information retrieval (PIR) [11], [12] between the client and the broker, thus implementing a private distribution of the ads delivered. However, the PIR scheme used was impractical for a real-time use and made it impossible to retrieve ads on-the-fly.

Anonymous browsing solutions such as TOR [13] can perfectly hide the client's identity which satisfies the privacy requirements. But such kind of implementations make it more difficult for click-fraud and potential collusion detection.

Hardware design can be another effective solution. In [6], the design was based on the usage of secure hardware-based PIR which guaranteed strong privacy. But the evolvement of a secure coprocessor was a very strong assumption which cannot be satisfied in many situations.

Guha *et al.* proposed *Privad* which was a complete system for privacy-preserving targeted advertising in [9], [14]. *Privad* introduced a *reference monitor* to watch the client software and to ensure that no data is sent by the client to the broker by a covert channel, a *dealer* worked as an anonymizing proxy between the client.

In [15], Toubiana *et al.* proposed *Adnostic*, which can extract and categorize the keywords locally by a Firefox extension. However, unlike *Privad*, *Adnostic* did not hide the clients' web browsing history from the broker, which made the privacy model quite weak. Moreover, *Adnostic* protected the client's privacy via transmitting a bunch of ads which inevitably increased network latency.

As a subset of targeted advertising, location-based advertising has attracted much attention in recent years. In [17], Guha *et al.* proposed *Koi*, a location-privacy platform for smartphone apps. *Koi* delivered ads to the client utilizing a matching protocol between different clients, thus keeping the client's location unknown to the broker. *Koi* can achieve excellent performance when there are a lot of users in a specific area, however, when users are sparsely distributed, the performance of *Koi* will be influenced significantly. In [18], Lu *et al.* introduced *PLAM*, a privacy-preserving framework for location-based service. *PLAM* was mainly based on cryptographic tools such as BV homomorphic encryption to achieve *k-anonymity* [19] and *l-diversity* [20]. However, the main drawback for *PLAM* is the same as that of *Koi*, which means that the client intensity greatly affects the system's performance. In [8], Pang *et al.* designed a privacy-preserving protocol for location-based advertising mainly based on PIR and homomorphic encryption. However, the scope of application of this method is quite limited to 2-dimensional location-based advertising, which makes it hard to apply to other scenarios.

VI. Conclusion

In this paper, we have a privacy-preserving protocol for keyword-based real-time advertising, called PROTA. By the highly efficient storage structure of the Bloom filter, we can allow the client to match the corresponding advertisers on the client side with acceptable computation and communication overheads. The ad exchange cooperates with a trusted third party to complete the bidding and delivering phase without privacy leakage. We have also designe a "e-coin" mechanism to ensure the correctness of the charging meanwhile preserving the privacy properties. Our evaluation results have shown that our protocol can perfectly protect the client's privacy with acceptable computation time and communication costs.

References

[1] B. H. Bloom, "Space/time trade-offs in hash coding with allwable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[2] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Theory of Cryptography*, pp. 325–341, Springer, 2005.

[3] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith, "Public key encryption that allows pir queries," in *CRYPTO '07*, pp. 50–67, Springer, 2007.

[4] K. Henry, *The Theory and Applications of Homomorphic Cryptography*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 2008.

[5] E. Teske, "Square-root algorithms for the discrete logarithm problem (a survey)," in *In Public Key Cryptography and Computational Number Theory, Walter de Gruyter*, pp. 283–301, 2001.

[6] M. Backes, A. Kate, M. Maffei, and K. Pecina, "Obliviad: Provably secure and practical online behavioral advertising," in *S& P '12'*, pp. 257–271, IEEE, 2012.

[7] A. Reznichenko, S. Guha, and P. Francis, "Auctions in do-not-track compliant internet advertising," in *CCS '11*, pp. 667–676, ACM, 2011.

[8] Y. Pang, Y. Chen, P. Liu, F. Qiu, F. Wu, and G. Chen, "Pola: A privacy-preserving protocol for location-based real-time advertising," in *IPCCC '14*, pp. 1–8, IEEE, 2014.

[9] S. Guha, B. Cheng, and P. Francis, "Privad: Practical privacy in online advertising," in *NSDI '11*, pp. 169–182, 2011.

[10] A. Juels, "Targeted advertising ... and privacy too," in *Topics in Cryptology–CT–RSA '01*, pp. 408–424, Springer, 2001.

[11] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *Journal of the ACM*, vol. 45, no. 6, pp. 965–981, 1998.

[12] C. Cachin, S. Micali, and M. Stadler, "Computationally private information retrieval with polylogarithmic communication," in *EUROCRYPT'99*, pp. 402–414, Springer, 1999.

[13] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *USENIX Security Symposium '04*, pp. 303–320, 2004.

[14] S. Guha, A. Reznichenko, K. Tang, H. Haddadi, and P. Francis, "Serving ads from localhost for performance, privacy, and profit," in *HotNets '09*, 2009.

[15] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas, "Adnostic: Privacy preserving targeted advertising," in *NDSS '10*, 2010.

[16] M. Götz and S. Nath, "Privacy-aware personalization for mobile advertising," in *CCS '12*, pp. 662–673, ACM, 2012.

[17] S. Guha, M. Jain, and N. Padmanabhan, Venkata, "Koi: A location-privacy platform for smartphone apps," in *NSDI '12*, pp. 14–14, 2012.

[18] R. Lu, X. Lin, Z. Shi, and J. Shao, "Plam: A prvacy-preserving framework for local-area mobile social networks," in *INFOCOM '14*, pp. 763–771, IEEE, 2014.

[19] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[20] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *TKDD '07*, vol. 1, 2007.