A Secure and Privacy-Preserving Targeted Ad-System

Elli Androulaki and Steven M. Bellovin
{elli, smb}@cs.columbia.edu

Columbia University

Technical Report CUCS-044-09

Abstract. Thanks to its low product-promotion cost and its efficiency, targeted online advertising has become very popular. Unfortunately, being profile-based, online advertising methods violate consumers' privacy, which has engendered resistance to the ads. However, protecting privacy through anonymity seems to encourage click-fraud. In this paper, we define consumer's privacy and present a privacy-preserving, targeted ad system (PPOAd) which is resistant towards click fraud. Our scheme is structured to provide financial incentives to to all entities involved.

1 Introduction

Thanks to its ability to target audiences combined with its low cost, online advertising has become very popular throughout the past decade. However, current profile-based advertising techniques raise privacy risks and may contravene users' expectations, while privacy-preserving techniques, e.g., anonymous browsing, create many opportunities for fraud. In this way, security and privacy seem to contradict each other. In this paper we show that the aforementioned concepts are not mutually exclusive. In particular, we analyze the privacy concerns raised by online advertising as well as the subsequent security issues, and propose a privacy preserving set of protocols that provide targeted ads with guaranteed fraud detection.

Privacy Concern: Targeted Ads To increase their banner-ads' effectiveness, *publishers* — usually service oriented websites paid to show advertising spots of other companies' products — choose their ads based on users' browsing activity. More specifically, third party cookies enable special ad networks to track users' browsing activity across multiple websites, construct very accurate user-profiles [KW06], and target ads accordingly. These advertising models track users even on sensitive sites, such as medical information websites, which could result in embarrassing advertisements appearing on other sites and in other contexts. A recent study [TKH+09] show broad rejection of the concept:

> Contrary to what many marketers claim, most adult Americans (66%) do not want marketers to tailor advertisements to their interests. Moreover, when Americans are informed of three common ways that marketers gather data about people in order to tailor ads, even higher percentages–between 73% and 86%—say they would not want such advertising.

The study found that over half of Americans felt that the punishment for illegal use of personal information should be jail time for the executives or that the company "be put out of business". The privacy issues become more serious when a conversion takes place, i.e., an online credit-card-based purchase or any activity which requires a login, thus linking a profile to a particular identity.

Security Concerns: Fraudulent Clicks In the mechanism described before, publishers and ad-networks get paid by the advertisers in proportion to the number of clicks an advertisment receives from users. To dishonestly increase their revenue, publishers often fake clicks on ads. The existing privacy-preserving techniques, such as anonymizing networks, make detection of fraudulent clicks more difficult as all user identification elements are concealed.

Our Contribution In this paper we present an online target advertising technique combining both privacy and security, PPOAd. More specifically,

1. we provide a concrete definion of consumers' privacy
2. we present a privacy-preserving mechanism for the current ad-system infrastructure guaranteeing similar or better revenues for all the entities involved
3. we present a privacy-preserving mechanism for click-fraud detection and show how this mechanism is applied in our system, and
4. we based our protocols on ecash and unlinkable credential systems

Organization In the following section we present current ad-systems' architecture. In sections 3 and 4 we demonstrate our system's requirements, threat model and protocols, while in sections 5 and 6, we elaborate on our system's security, privacy and innovation w.r.t. the exising work.

2 Targeted-Ads System Architecture

Except for **users** — the online consumers — in a typical advertising mechanism, the principle parties are **advertisers**, **ad networks** and the **publishers**. **Advertisers** are the companies selling and promoting a particular product or group of products. **Publishers** are usually service-oriented websites paid to *publish* advertisements of advertisers' products. **Ad networks** are paid by advertisers to choose the list of advertisements which will appear on publishers and filter the clicks the ads receive. Typical examples of ad-networks are Doubleclick (owned by Google), Atlas Solutions (owned by Microsoft), Brightcove, and more. It is often the case that an ad network offers various services and also acts as a publisher.

When a user visits a website(publisher), the browser sends to the publisher some pieces of information called cookies, which link multiple visits of the same user. In fact, a special type of cookies, the *third party cookies*, are sent during the publishers' visit to the corresponding ad networks, who can now trace user activity across multiple websites. In this way, especially as ad networks collaborate with many publishers, they construct very accurate user profiles and target ads accordingly. There are many policies regarding how ad-networks and publishers are paid. The most popular one is the "cost per click" (CPC), where both parties are paid by the advertisers in proportion to the number of clicks the latters' ads receive.

As clearly shown before, targeted ads violate privacy, while CPC payment method motivates many attacks: publishers may fake clicks on ads they publish to increase their income, while advertisers may generate clicks on their competitors' advertisements to deplete the latter's daily advertising badget. Detection of click-fraud is currently the responsibility of ad networks. Unfortunately, it is apparent that any conventional mechanism concealing users' browsing activity may strenghten click fraud.

3 Requirements-Threat Model

In this section we will define *privacy*, *security* and *deployability* in the context of our system w.r.t. our system's requirements and threat model.

3.1 Requirements

Application layer *privacy* and *security* are the core requirements in our system. Privacy refers to the user-protection, while security refers to the protection of the other entities of the system. More specifically, we define privacy, as the union of:

- *User Activity Unlinkability.* No system entity should be able to profile a particular honest user, i.e., link two or more web activities as having originated by the same party, and
- *User Anonymity.* No system entity should be able to link a particular browsing activity to an identity.

In addition, we define security as the combination of the following properties:

- *Correctness.* We require that if all parties are honest, advertisers will pay publishers and ad networks in accordance to the number of clicks their ads have received, while privacy is maintained.

– *Fairness.* We require that parties in our system will be paid if and only if they do their duty properly.
– *Accountability.* Our system should also be accountable, i.e., misbehaving parties should be detected and identified.
– *Unframability.* We require that no user can frame an honest user for being responsible for a misbehavior, i.e., for click-fraud. It is conceivable that strong accountability implies unframability.
– *Mis-Authentication.* Unless authorized, no user should be able to make use of our system.

We can easily see how the click fraud detection requirement is covered through the fairness and accountability requirements: fairness requires that publishers should not receive payments for fake clicks on a particular advertisement, while accountability requires that the attacker is traced.

In addition, we require that our system provide *similar ad-efficiency*, which would result in similar profitability to the parties involved. At least as important, it must be deployable. Similar ad-efficiency and, thus, *similar profitability* for publishers and ad networks aims to eliminate any monetary constraints against the adoption of a new system. *Deployability* is important for the same reasons. We examine deployability from three aspects: (a) w.r.t. our system's architecture: not substantial changes in current ad-system architecture should be required for our protocols to be applied; (b) w.r.t. our threat model, as we will describe later on; (c) w.r.t. to computation needs.

It is essential to note that both privacy and security provisions are required in the application layer. Also, we extend the current ad-system architecture with a single entity — which may or may not be distributed — the User Ad Proxy (UAP), which acts as a mediator between the user and each visiting website.

3.2 Threat Model

Ad-systems' strong monetary nature, imposes "following the money" the safest way to define our adversaries' motives and powers. In what follows, we examine our adversary w.r.t. users' privacy and ad-system's security.

Publishers may be "curious" w.r.t. users' privacy, i.e., they may collaborate with ad networks, advertisers or other users in order to reveal the identity of a particular user or to link browsing activities of the same user. In addition, we assume that publishers are "honest and dishonest" w.r.t. the ad networks and advertisers. In particular, we assume that they do provide correct user-profile related information to the ad networks, but may attempt to fake clicks to the advertisements they publish in order to increase their revenues.

Ad networks' revenues depend on the efficiency of the way they list ads in the various publishers, as well as on their credibility. Ads' efficiency depends on the accuracy of users-profiling, while credibility depends of the ad network's click frauds' detectability. It is, consequently, reasonable to assume that ad networks are "honest but curious", w.r.t. users, while they are "honest" w.r.t. advertisers.

Advertisers are considered to be "curious" w.r.t. the users. In particular, since advertisers have no direct interaction with them, we believe that they may collaborate with publishers or ad-networks to make user-profiling more accurate.

UAP is considered to be "honest but curious" w.r.t. the users. More specifically, we assume that is trusted to perform its functional operations honestly towards the users, but may collaborate with publishers or any other entity to link separate browsing activities of the same user. We also adopt a economic model so that UAP does not have a motive to cheat the advertisers.

4 A Privacy preserving Targeted-Ad System

As mentioned in the previous section, we extend the current ad-system architecture with the User Ad Proxy(UAP). UAP may be considered either as a single entity or as a group of collaborating entities and acts as a communication mediator between a user U visiting a publisher-website Pub and Pub. It is important to note that to hide any lower layer information emitted, U interacts with the rest of the system entities through an anonymizing network, while to automatically erase any cookies acquired and to be able to communicate with UAP or an UAP-member (if distributed), user-side installs a piece of software, which basically establishes an anonymous — communication layer — registration of user with the UAP.

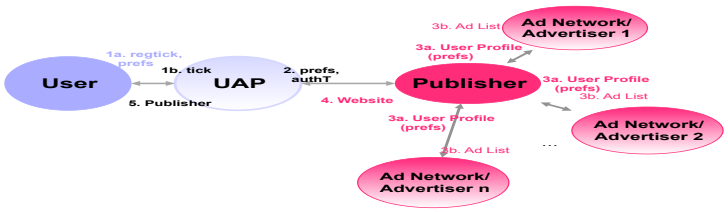

Fig. 1. Visiting a Website.

The three core operations of our system: *(a)* the registration procedure of a user U at PPOAd, during which U obtains credentials to use the services of UAP, *(b)* the visit to a publisher, where a PPOAd-user requests a webpage (fig. 1), and *(c)* the ad-clicking procedure, where the user clicks on one of the publisher's ads (fig. 2). For convenience, we will assume that a user U is interacting with a publisher Pub. In addition, we will assume a single UAP, while in section 5, we will refer to the distributed UAP case.

Our scheme is based on the use of two types of tokens, issued by the user-UAP collaboration during the registration procedure: a registration credential regtick, which may authorize U as member of PPOAd multiple times anonymously and unlinkably, and a wallet with adticks, W_{adtick}, which will enable U to click on ads. regticks are blind towards the UAP, their possession can be demonstrated by their owner anonymously and unlinkably many times, each time resulting in a session-oriented ticket tick. Issued by the valid collaboration between U and the UAP, adticks are blind towards the UAP and can only be used for a limited number of times (MaxClicks) strictly by the person who issued them. For security purposes, U's identity is revealed in the following two cases: *(a)* when U attempts to make use of the same adtick more than once, or *(b)* if more than MaxClicks adticks of U are used for the same ad. We make use of regticks to achieve privacy w.r.t. UAP and adticks to achieve privacy and security w.r.t. to all the entities. Both tokens have an expiration date, so that users need to update their subscription on a monthly basis. In this way, we avoid uneccessary computations on a regular basis, as misbehaving parties can be detected and removed from the system.

As shown in fig.1, when requesting for a webpage, U sends to UAP his ad-preferences, demonstrates knowledge of his regtick and proves that his regtick is not among the black-listed ones. UAP contacts the website and provides it with the U-specified ad-preferences. Ads are then shown to U accordingly.

When U clicks on an ad (see fig.2), he uses one of the adticks he has obtained at the registration proocedure. The adtick is then linked to the following combination:

{publisher || ad network || product-serial},

which identifies the particular ad. If U clicks intentionally on the same ad more than a pre-defined number of times using his adtick s, he will risk his privacy, as his identity will be revealed. However, U can choose instead to open an account for clicking on that particular ad, which will enable the ad network to decide whether series of U's clicks on that ad are legal or fraudulent. If classified as malicious, U's membership credential will be blacklisted.

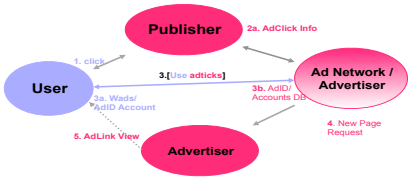

Fig. 2. Clicking on Ads.

As we will see later on, we use the blacklistable version of unlinkable credential system of [TAKS07] for the registration credentials regticks and the accountable ecash [JCL06] scheme for adticks.

In what follows, we will elaborate on the building blocks we used to construct our protocols and based on them we will proceed with the detailed description of our scheme.

4.1 Building Blocks

In this section, we present our primitives: group, blind signature schemes, blacklistable anonymous credential systems as well as digital cash systems.

Ecash An E-Cash [CHL05][JCL06] system consinsts of three types of players: the *bank*, *users* and *merchants*. The input and output specifications of the basic operations are as follows. For convenience, we will assume that the operations take place between a merchant M, a user U and the Bank B.

- $(pk_{\mathsf{B}}, sk_{\mathsf{B}}) \leftarrow \mathsf{EC.BKeyGen}(1^k, params)$ and $(pk_{\mathsf{U}}, sk_{\mathsf{U}}) \leftarrow \mathsf{EC.UKeyGen}(1^k, params)$, which are the key generation algorithm for the bank and the users respectively.
- $\langle W, \top \rangle \leftarrow \mathsf{EC.Withdraw}(pk_{\mathsf{B}}, pk_{\mathsf{U}}, n)$ $[\mathsf{U}(sk_{\mathsf{U}}), \mathsf{B}(sk_{\mathsf{B}})]$. In this interactive procedure, U withdraws a wallet W of n coins from B.
- $\langle W', (S, \pi) \rangle \leftarrow \mathsf{EC.Spend}(pk_{\mathsf{M}}, pk_{\mathsf{B}}, n)$ $[\mathsf{U}(W), \mathsf{M}(sk_{\mathsf{M}})]$. In this interactive procedure, U spends a digital coin with serial S from his wallet W to M. When the procedure is over, W is reduced to W', M obtains as output a coin (S, π), where π is a proof of a valid coin with a serial number S.
- $\langle \top/\bot, L' \rangle \leftarrow \mathsf{EC.Deposit}(pk_{\mathsf{M}}, pk_{\mathsf{B}})$ $[\mathsf{M}(sk_{\mathsf{M}}, S, \pi), \mathsf{B}(sk_{\mathsf{B}}, L)]$. In this interactive procedure, M deposits a coin (S, π) into its account in the bank. If this procedure is successful, M's output will be $\top$ and the bank's list L of the spent coins will be updated to L'.
- $(pk_{\mathsf{U}}, \Pi_G) \leftarrow \mathsf{EC.Identify}(params, S, \pi_1, \pi_2)$. When the bank receives the two coins with the same serial number S and validity proofs π_1 and π_2, it executes this procedure, to reveal the public key of the violator accompanied with a violation proof Π_G.
- $\top/\bot \leftarrow \mathsf{EC.VerifyGuilt}(params, S, pk_{\mathsf{U}}, \Pi_G)$. This algorithm, given Π_G publicly verifies the violation of pk_{U}.
- $\{(S_i, \Pi_i)\}_i \leftarrow \mathsf{EC.Trace}(params, S, pk_{\mathsf{U}}, \Pi_G, D, n)$. This algorithm provides the list of serials S_i of the ecoins a violator pk_{U} has issued, with the corresponding ownership proofs Π_i.
- $\top/\bot \leftarrow \mathsf{EC.VerifyOwnership}(params, S, \Pi, pk_U, n)$. This algorithm allows to publicly verify the proof Π that a coin with serial number S belongs to a user with public key pk_U.

[JCL06] is a money-laundering prevention version of [CHL05], where anonymity is revoked when the spender spends more coins to the same merchant than a spending limit. In this case ecoins are upgraded to $C = (S, V, \pi)$, where V is a merchant-related locator, while EC.Identify and EC.VerifyGuilt procedures are upgraded to the DetectViolator and VerifyViolation to support the extended violation definition.
Security Properties: (a) Correctness. (b) Balance. No collection of users and merchants can ever spend more coins than they withdrew. *(c) Identification of Violators.* Given a violation and the corresponding proofs of guilt, the violator's public pk_{U} key is revealed such that EC.VerifyViolation accepts. *(d) Anonymity of users.* The bank, even when cooperating with any collection of malicious users and merchants, cannot learn anything about a user's spendings other than what is available from side information from the environment. *(e) Exculpability.* An honest user U cannot be accused for conducting a violation such that EC.VerifyViolation accepts. (f) Violators' Traceability. Given a violator Uwith a proof of violation Π_G, this property guarantees that EC.Trace will output the serial numbers of all coins that belong to U along with the corresponding proofs of ownership, such that for each one of them VerifyOwnership accepts.

Blacklistable Anonymous Credentials (BLAC) The entities in the blacklistable credential system BLAC of [TAKS07] are the Group Manager GM, a set of service providers SPs and users. The procedures supported are the following:

- $\langle \mathsf{gpk}, \mathsf{gsk} \rangle \leftarrow \mathsf{BLAC.Setup}[\mathsf{GM}(1^k)]$. This algorithm generates a group public key gpk and the GM's secret group information gsk .
- $\langle \mathsf{cred}_{\mathsf{U}}, \mathsf{JLog}_{\mathsf{U}} \rangle \leftarrow \mathsf{BLAC.Register}(\mathsf{gpk})[\mathsf{U}, \mathsf{GM}(\mathsf{gsk})]$. When this interactive registration ends, U has obtained his membership credential $\mathsf{cred}_{\mathsf{U}}$.
- $\langle \top/\bot \rangle \leftarrow \mathsf{BLAC.Authenticate}(\mathsf{gpk})$ $[\mathsf{U}(\mathsf{cred}_{\mathsf{U}}), \mathsf{SP}(\mathrm{BL})]$. In this interactive procedure, U proves to SP that he is a valid (non-blacklisted) member of the group.
- $\langle BL' \rangle \leftarrow \mathsf{BLAC.BLAdd}[\mathsf{SP}(BL)]$, where a service provider ads a credential (ticket) to the blacklist BL.
- $\langle tick \rangle \mathsf{BLAC.BLExtract}[\mathsf{SP}(BL)]$, where SP extracts an element from the blacklist.

- $\langle BL' \rangle \leftarrow \mathsf{BLAC.BLRemove}[\mathsf{SP}(BL)]$, where SP removes a credential from the blacklist.

Security Properties: (a) Correctness. (b) Mis-authentication Resistance. No unregistered user or collection of unregistered users should be able to authenticate themselves. *(c) Blacklistability.* SPs may blacklist any misbehaving user of the system and restrict him from any ability of authenticating himself. *(d) Anonymity.* SPs may only learn whether a user is blacklisted or not; no identification information may be leaked. *(e) Non-framability.* An honest user should never be blocked from access.

Group Signature Schemes (GSS) In a typical GSS, there is a group manager (GM), the group-members, who act as signers (let each be S) and produce signatures on behalf of the group. The procedures supported are the following:

- $(\mathsf{gpk}, \mathsf{gsk}) \leftarrow \mathsf{GS.Setup}(1^k)$. This algorithm generates a group public key gpk and the GM's secret group information gsk .
- $\langle \mathsf{bgusk_S}, \mathsf{JLog_S} \rangle \leftarrow \mathsf{GS.Join}(\mathsf{gpk})[\mathsf{S}, \mathsf{GM}(\mathsf{gsk})]$. When this interactive join procedure ends, an S obtains a secret signing key $\mathsf{bgusk_S}$, and the GM (group manager) logs the join transcript in the database D.
- $\sigma \leftarrow \mathsf{GS.Sign}(\mathsf{gpk}, \mathsf{bgusk_S}, m)$. This algorithm generates a group signature on a message m.
- $\langle \top / \bot \rangle \leftarrow \mathsf{GS.Verify}(\mathsf{gpk}, m, \sigma)$. This is a verification algorithm.
- $\mathsf{Ms} \leftarrow \mathsf{GS.Open}(\mathsf{gsk}, \sigma, D)$. With this algorithm the GM determines the identity of the group member who generated the signature σ.

Security Properties: (a) Anonymity. Given a signature and two members, one of whom is the originator, the adversary can identify its originator among the group members no better than randomly. *(b) Unforgeability.* The adversary cannot produce a valid group signature without owning group membership information. *(c) Non-framability.* The adversary cannot create a valid group signature that opens to another group member.

4.2 The PPOAd Protocol in detail

Setup. All entities run a key generator algorithm (EC.UKeyGen) to generate their signature key-pairs and publish their public information.

UAP runs EC.BKeyGen twice to establish the two accountable ecash schemes, which will be used for adticks(see, section 4.1). In addition to its keys' generation, UAP runs the BLAC.Setup procedure of the blacklistable anonymous credential (BLAC) scheme (see, section 4.1) for user-registration purposes, while it maintains two blacklists: the TempBL, where it stores the credentials in question, and, the PermBL, which is the official blacklist of the system.

Each ad network AdNet runs GS.Setup to generate the administration information for the group of publishers $\mathsf{G}^{\mathsf{AdNet}}$ it provides ads with: $\{\mathsf{gpk}^{\mathsf{AdNet}}, \mathsf{gsk}^{\mathsf{AdNet}}\}$. In response, each publisher collaborating with AdNet, runs GS.Join with AdNet to obtain membership in $\mathbb{G}^{\mathsf{AdNet}}$.

Registration (PPOAd.Register) This is the case where a user U registers to UAP such that the former makes use of PPOAd's privacy services. PPOAd.Register consists of the following steps:

1. U provides the UAP with a piece of identification information. This can be a credit card, which will be used to pay U's subscription. U runs EC.UKeyGen to issue a signature key pair ($\mathsf{pk_U}$, $\mathsf{sk_U}$), with he will be identified in the PPOAd system.
2. U and UAP collaborate in a BLAC.Register procedure, where U's credential $\mathsf{regtick_U}$ is issued. $\mathsf{regtick_U}$ is blind towards UAP.
3. U and UAP collaborate in a BLAC.Authenticate procedure, so that UAP obtains a transcript of the $\mathsf{regtick_U}$ authentication phase, $\mathsf{mem_U}$. Note that the $\mathsf{mem_U}$ which was obtained by UAP in this way, serves blacklistability pursposes and cannot be linked to later authentications of U through $\mathsf{regtick_U}$.
4. U and UAP collaborate in two EC.Withdraw procedure, for the former to obtain two wallets $\mathsf{W}^{\mathsf{f,l}}_{\mathsf{ads}}$ of accountable ecash each corresponding to the two different settings of accountable ecash established in the setup phase.
5. UAP stores in its membership database the new user's entry: $\{\mathsf{U}, \mathsf{pk_U}, \mathsf{mem_U}\}$, while it provides U with a signed proof of payment: $\mathsf{PaymRec} = \mathsf{Sig_{UAP}}(timestamp, \mathsf{U})$.

In what follows, we will assume that a user U visits a website Pub, which is in contract with a number of ad networks $\mathsf{Adv}_1, \ldots, \mathsf{Adv}_m$, who provide the website with ads.

Ad-targeting (PPOAd.Target) This procedure involves the ad-targeting taking place when U visits Pub.

1. *User Authorization*: U interacts with UAP in a BLAC.Authenticate procedure to authenticate himself as a non-blacklisted member of the PPOAd system. In this procedure U demonstrates knowledge — in a zero knowledge fashion — of his membership credential $\mathsf{regtick}_\mathsf{U}$. Let AuthT be the corresponding transcript of U-UAP interactions.
2. tick *issue phase*: UAP issues a signed, dated permission tick, which will enable U to access the website requested. tick may have the form of
$$\mathsf{tick} = \mathsf{Sig}_{\mathsf{UAP}}(timestamp, \mathsf{AuthT}).$$
3. *Preferences setup.* In this phase, U sets up his ad-preferences and sends them to the UAP. UAP then sends the webpage http request with U's preferences. As we will see later, the preferences-related info provided to UAP does not enable U activity-tracking neither by the UAP, nor by the requested website.
4. *Targeting.* Ad networks, who receive U's preferences as coming from UAP itself, process the ad-preferences and provide Pub with the corresponding list of ads.
5. Pub *Visit.* U provides tick to Pub and the Pub-webpage is presented to U.

Ad-clicking (PPOAd.Adclick) This operation refers to the case, where U has already visited Pub and clicks on one of the ads an ad network $\mathsf{AdNet_i}$ provided to Pub. The series of interactions involve the following:

1. *Clicked Ad's website request.* Pub sends the ad-click information to the clicked ad's website, which is essentially one of the advertisers in contract with $\mathsf{AdNet_i}$. Let $\mathsf{Adv_j}$ be the one. The ad-click information includes $\mathsf{AdNet_i}$, Pub, AuthT and a timestamp. Note that this step is currently performed in ad-systems and serves billing and user-profiling purposes. Note that AuthT can be considered as a session identifier for the U.
2. AdID *construction.* In this phase the ad network, advertiser and clicked-product's identifier is popularized to U. The complete ad's identity would then be the following:
$$\mathsf{AdID} = \{\mathsf{Pub}||\mathsf{AdNet_i}||\text{product ID}\},$$
where we assume that the same products of different advertisers have different identification numbers. As we will discuss in section 5, in addition to the AdID, an AdID-related key-pair is constructed $(pk_{\mathsf{AdID}}, sk_{\mathsf{AdID}})$.
3. adtick-*based Authorization.* Let MaxClicks be the number of times an honest user usually clicks on an ad.[1] Based on how many times U has — over all his browsing activity — clicked on that particular AdID, we have three adclick protocols of $\mathsf{U\text{-}AdNet_i}$ interaction:
   (a) If U has clicked on the same AdID fewer than MaxClicks times, he and $\mathsf{AdNet_i}$ collaborate in EC.Spend procedure, so that U spends one of his $\mathsf{W}^{\mathsf{i}}_{\mathsf{ads}}$ digital coins to the AdID related key-pair.
   (b) If U has clicked exactly MaxClicks times to the same AdID, he and $\mathsf{AdNet_i}$ commit in a similar (as before) EC.Spend procedure for one of the coins of U's $\mathsf{W}^{\mathsf{l}}_{\mathsf{ads}}$ wallet. In addition, U and $\mathsf{AdNet_i}$ collaborate in an EC.KeyGen procedure for U to create an account $(pk_{\mathsf{U}}^{\mathsf{AdID}}, sk_{\mathsf{U}}^{\mathsf{AdID}})$ within $\mathsf{AdNet_i}$ for that particular AdID. $\mathsf{AdNet_i}$ stores $\{\mathsf{pk_U}\mathsf{AdID}, \mathsf{AuthT}, \mathsf{AdID}\}$ to its database.
   (c) If U has clicked on the same AdID more than MaxClicks times, he has already been issued an AdID-account. Thus, he demonstrates knowledge of $sk_{\mathsf{U}}^{\mathsf{AdID}}$. In this way, his behavior towards this AdID will be traceable.

   We can see that a user trying to attack an advertiser using PPOAd will eventually have his click-activity for that particular ad traced. In this way, $\mathsf{AdNet_i}$ may have all the information necessary to characterize the sequence of clicks on that AdID as malicious or benign. Different CPC rates may apply in this case.

[1] This number varies from two to four, depending on how interesting that product is, and should be defined after suitable research.

4. If everything is fine, the the Advj website is presented to U.

If at any point of the procedure, U declines to cooperate, $\mathsf{AdNet_i}$ or Pub report AuthT to UAP, who can then run BLACK.BLAdd on TempBL to put a temporary hold on $\mathsf{regtick_U}$ which corresponds to AuthT. Note that thanks to the properties of BLAC system adopted, UAP does not need to know the user U or his $\mathsf{regtick_U}$. On the other hand, if U tries to click on the same AdID more MaxClicks times using his adticks, he will need to spend more than MaxClicks coins of his $\mathsf{W^i_{ads}}$ wallet or more than one coin of his $\mathsf{W^l_{ads}}$ to the same AdID. Because of the accountable ecash properties, this will result in revocation of $\mathsf{pk_U}$, while $\mathsf{regtick_U}$ will be immediately blacklisted (through $\mathsf{mem_U}$).

Update Membership (PPOAd.UpdateMembership) To enforce payment of its registered members' monthly contribution, at the end of this prefixed period UAP changes its credentials' parameters. To continue making use of PPOAd's services, users contact UAP by providing the corresponding identification and payment (most recent PaymRec) information. Also, each user and UAP commit in a BLAC. Authenticate protocol, for the former to prove that his old credential is not among the blacklisted ones. If a user's $\mathsf{Uregtick_U}$ is not blacklisted, U pays his monthly contribution, issues a new $\mathsf{regtick_U}$ and receives new PaymRec. On the other hand, if $\mathsf{regtick_U}$ is blacklisted or U does not pay, his old credential will not be up to date and thus will not be possible for him to use PPOAd's services.

5 System Considerations

In this section, we will elaborate on the *security* and *privacy* properties of our system, as well as on other practical issues.

User-ad-preferences play an important role for our system's *ad-efficiency*. As mentioned in previous section, after his registration to the PPOAd (PPOAd.Register) the user obtains and installs software to handle the PPOAd's interactions, which also includes a user-preferences-related section (as the one in [CDFP03]). Depending on how targeted wishes his ads to be, the user creates many partial profiles by choosing various types of products he is interested in and the particular products in each category individually that may be of his interest. When the user visits a website and after the PPOAd.Authenticate phase — the user-software obtains the classification of the requested website and forwards to UAP the corresponding partial profile. Assuming that the lower layer information, i.e., consumer's machine's IP is hidden towards the UAP, because of the BLAC system unlinkability property, the latter cannot link the partial profiles of the same user. In addition, being partial (related to the website), while prone to change at any time, the same partial profile may commonly be met across different users and will not be enough to link browsing activities of the same consumer across different websites.

AdID **key-pair construction**. To preserve security and privacy, it is essential that each AdID-combination: {Pub, AdNet, productID}, where Pub is the visited website, AdNet is the ad network providing the ad and productID the serial of the product, is assigned a different key-pair. To achieve this, AdNet constructs AdID's key-pair by contributing to the key-generation algorithm a pre-specified hash of the following quantity: $\mathsf{gpk_{Pub}}||pk_{\mathsf{AdNet}}||\mathsf{productID}$, where $\mathsf{gpk_{Pub}}$ is the Pub's public information in the $\mathbb{G}^{\mathsf{AdNet}}$, and pk_{AdNet} the public key of AdNet. In this way, the same key will be generated for the same AdID, without the need of precalculating it, while the probability that the same key is generated for two AdIDs is negligible.

Distributed UAP. If PPOAd is intended for large-scale use, it is important that UAP's functionality is distributed. This can be achieved through the use of blind group and group signatures, where UAP-related blind and plain signatures were used. In addition, depending on our privacy and computation efficiency requirements, we may group UAPs serving users of the same geographical area together. In this way, operations such as validity checks will be accelerated.

Privacy. Assuming that the partial profiles reveal nothing w.r.t. the consumer, and that user-cookies are succesfully erased through the PPOAd-software installed, privacy in our system is guaranteed through the ecash and BLAC systems' security properties (see, section 4.1 and 4.1). In particular, consumers' anonymity and activity unlinkability is provided directly via the anonymity and unlinkability properties of the blacklistable anonymous credentials used in PPOAd.Register and PPOAd.Target procedures and anonymity and unlinkability properties of the ecash schemes used in PPOAd.Adclick procedure. Note that even when a consumer clicks on the same AdID more than MaxClicks times, the former's behavior towards that particular AdID is only traceable (when and how often the consumer clicks on it) and not his overall browsing activity.

Security. Each part of PPOAd's security is satisfied. Correctness is guaranteed through the correctness of the schemes adopted. Mis-authentication resistance is achieved through the corresponding property of the blacklistable anonymous credential system used at the authorization phase of our protocols. Unframability is guaranteed through the combination of the mis-authentication resistance property and the ecash nature of the adticks: being unforgeable and ecash-based, only an authenticated PPOAd-user who issued the adticks can use them successfully. Fairness and accountability are achieved also through the accountable ecash security properties (see, section 4.1): a user trying to click at the same ad many times will either have his public key revealed or his click-activity w.r.t. that ad traced. If the latter is the case, and the user is classified as malicious, he will be automatically be blacklisted and his ad-clicks ignored.

A Market Model. Threats towards security continue to exist. In our threat model, we assumed that UAP restricts users to a single registration to PPOAd. However, it is conceivable that in the real world, corrupted UAP entities — since they are paid by the users — may be tempted to issue multiple accounts to the same party, which would enable the latter to forge clicks without limit. It is thus critical that we offer monetary incentive for the system entities to behave according to our threat model. In addition, since targeted advertising is already very profitable, we need to create incentives to direct it as we propose. In this section we elaborate on a market model towards for both cases.

In response to our threat model issue, we require that UAP entities are paid by both, the user — through his PPOAd-subscription — and by the ad networks who also benefit from click-fraud. Wanting to maintain their clientele, UAP will be forced to be "honest" in their functional operations towards both entities: users and ad networks.

As far as the PPOAd application is concerned, ad networks already have incentives to participate in our system: through the PPOAd click-fraud detection mechanism, ad networks' fraud detection ability — thus their credibility — will be enhanced. In addition, despite our privacy provisions, ad networks may still target ads even more effectively: the targeting procedure is now based on partial profiles provided by the user himself, while it is likely that their audience is extended with users who — strictly for privacy reasons — had so far removed ads from their browsers. Being offered better click-fraud detection rates, advertisers would also benefit from PPOAd.

6 Related Work

Fraudulent Click detection has been attempted in the past. In particular, Jakobsson, MacKenzie and Stern in [JMS99] introduce an ad system where advertisers (in their system are called merchants) utilize e-coupons to detect malicious actions. However, in their scheme they do not deal with privacy the same way we defined it.

Combining targeted ads and privacy has been attempted in the past. Juels in [J01] has suggested a target ad technique with the use of third parties, the nogotiants, which would update a bulletin board with users' ad-preferences. Although perfectly secure in terms of privacy, they do not deal with our second security consern. Claessens and Diaz in [CDFP03] in fact suggested a more lightweight privacy preserving target ad system, where users would be grouped in terms of profiles for them to be presented with ads. V. Toubiana et al. in [TNB+09] have transfered the targeting mechanism to a browser extension, in a private way towards ad networks and publishers. However, though there are some suggestions, they do not consider click fraud. To

privately target ads, iPrivacy [SS01] ecommerce system, had their clients obtain anonymous email accounts — held by the company itslelf or the banks — bound to specific advertising profiles; in this way users only receive ads of their interests.

7 Conclusion

In this paper we have addressed privacy and security in the area of online targeted advertising. In particular, we provided a set of protocols providing targeted ads with similar efficiency as current systems and in a privacy preserving way. At the same time, the privacy provided in our system is conditional, guaranteed only for honest users.

Acknowledgments

We'd like to thank Chris Jay Hoofnagle and Christopher Soghoian for their comments on this paper.

References

[CDFP03] J. Claessens, C. Diaz, R. Faustinelli, and B. Preneel. Secure and privacy-preserving web banner system for targeted advertising, 2003. COSIC internal report, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.15.4382.

[CHL05] J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 302–321. Springer-Verlag, 2005.

[J01] A. Juels. Targeted advertising ... and privacy too. In *CT-RSA 2001: Proceedings of the 2001 Conference on Topics in Cryptology*, pages 408–424, London, UK, 2001. Springer-Verlag.

[JCL06] S. H. Jan Camenisch and A. Lysyanskaya. Balancing accountability and privacy using e-cash (extended abstract). In *Security and Cryptography for Networks*, 2006.

[JMS99] M. Jakobsson, P. D. MacKenzie, and J. P. Stern. Secure and lightweight advertising on the Web. *Computer Networks (Amsterdam, Netherlands)*, 31(11–16):1101–1109, 1999.

[KW06] B. Krishnamurthy and C. E. Wills. Generating a privacy footprint on the internet. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 65–70, New York, NY, USA, 2006. ACM.

[SS01] S. J. Stolfo and J. M. Smith. Method and system for user defined filtering of communications to anonymous users in a computer network, 2001. U.S. patent WO/2001/065442.

[TAKS07] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith. Blacklistable anonymous credentials: blocking misbehaving users without ttps. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 72–81, New York, NY, USA, 2007. ACM.

[TKH+09] J. Turow, J. King, C. J. Hoofnagle, A. Bleakley, and M. Hennessy. Americans reject tailored advertising and three activities that enable it, September 2009. http://ssrn.com/abstract=1478214.

[TNB+09] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and B. Solon. Priveads: Privacy preserving targeted advertising, 2009. http://crypto.stanforf.edu/privads/.