Auctions in Do-Not-Track Compliant Internet Advertising

Alexey Reznichenko
MPI-SWS
areznich@mpi-sws.org

Saikat Guha
Microsoft Research India
saikat@microsoft.com

Paul Francis
MPI-SWS
francis@mpi-sws.org

ABSTRACT

Online tracking of users in support of behavioral advertising is widespread. Several researchers have proposed non-tracking online advertising systems that go well beyond the requirements of the Do-Not-Track initiative launched by the US Federal Trace Commission (FTC). The primary goal of these systems is to allow for behaviorally targeted advertising without revealing user behavior (clickstreams) or user profiles to the ad network. Although these designs purport to be practical solutions, none of them adequately consider the role of the ad auctions, which today are central to the operation of online advertising systems. This paper looks at the problem of running auctions that leverage user profiles for ad ranking while keeping the user profile private. We define the problem, broadly explore the solution space, and discuss the pros and cons of these solutions. We analyze the performance of our solutions using data from Microsoft Bing advertising auctions. We conclude that, while none of our auctions are ideal in all respects, they are adequate and practical solutions.

Categories and Subject Descriptors

K.4.1 [**Public Policy Issues**]: Privacy

General Terms

Design, Economics, Human Factors

Keywords

Auctions, Online Advertising, Privacy, Targeting

1. INTRODUCTION

Third-party tracking of users is widespread and increasing [17]. One of the primary purposes of user tracking is behavioral advertising. Although many companies that participate in tracking and behavioral profiling claim to not gather Personally Identifying Information (PII), it is often easy to link tracking information with PII [16]. Concern over tracking continues to grow, and has led for instance to the Do-Not-Track initiative launched by the Federal Trace Commission (FTC), and proposals for a do-not-track registry [23]. Do-not-track, however, requires a public that is largely unaware of tracking to opt-out. It also forces a trade-off between tracking and behavioral targeting: a compromise that industry is not prepared to make. Not surprisingly, do-not-track has not gained a lot of traction with the public, and ad networks are actively fighting it.

Several recent research projects take a different tack (Adnostic [24], Nurikabe [18], and Privad [13]). Rather than simply opting out of tracking, or pushing for data-protection after the fact, they propose alternative advertising technologies that allow for behavioral targeting without requiring tracking. These are meant to be practical alternatives that industry finds attractive. None of these systems, however, adequately explore how to operate the auctions that are critical to current advertising systems. Without this component, these systems leave unanswered what revenue the *broker* (i.e. an ad network like Google) can earn, thereby reducing the likelihood that a non-tracking[1] advertising system will be of commercial interest. In this paper, we look at the problem of running auctions that leverage a *user profile* for ad ranking while keeping the user profile private.

Although the non-tracking advertising systems cited above differ in significant ways, they all share several key design components. All systems propose a *software agent* that runs at the client and generates a user profile. All systems at least share the privacy goal that this profile not be revealed. All these designs propose that the broker transmit multiple ads to the client, not all of which match the user profile. For instance, the ads may all be within a given interest category. The client then locally selects from among these ads which best match the user profile, and display these to the user. The client reports the result of this selection anonymously, and without letting the broker link together different components of the user profile. The key privacy mechanisms are therefore anonymity and unlinkability.

What does this have to do with auctions? The most common pricing model for online advertising systems today is Pay Per Click (PPC): the advertiser does not pay the broker for showing an ad to a user, rather it pays only if the user clicks on an ad. The broker selects which ads to show through an auction whereby advertisers bid against each other. In a PPC system, the broker maximizes revenue by ranking the competing ads according to

[1]By non-tracking, we mean no third-party tracking.

the $Bid \times ClickProbability$ product, and transmitting the highest ranking ads to the client where they are displayed in rank order. Of course, the broker doesn't know the precise click probability for every ad. Rather, the broker tries to predict the click probability as best it can. This prediction is based on a number of inter-related factors such as the ad keywords, the landing page keywords, the user search terms or keywords associated with the web-page being browsed, stored user characteristics, and so on.

The user profile has a strong effect on click probability. To give a simple example, say a user searches for "running shoe". Whether the user is a man or a woman, or prefers brand-name products or discount products, plays an important role in which running shoe ad he or she is more likely to click on. In a do-not-track compliant advertising system, the broker does not know the user profile: if the auction takes place at the broker in the same way that it does today, then the user profile will not be factored into the result. Therefore the highest $Bid \times ClickProbability$ ads won't be selected, leading to less revenue than should otherwise be possible.

This paper characterizes the problem, and proposes three basic solutions, one of which is a variant of a method proposed in [12]. Following the lead of the initial work on non-tracking advertising, this paper takes a pragmatic approach to the problem. It looks for a good trade-off between strict privacy guarantees and practical business and deployment concerns. As such, this paper explores the pros and cons of the three approaches in terms of not just privacy (both user and advertiser), but also revenue, overhead, and vulnerability to attack. It uses around 2TB of auction traces from Microsoft Bing to guide and validate the design choices.

Altogether, this paper makes the following contributions: 1) It proposes two new non-tracking auction designs, and a third based on the previous work but substantially improved. 2) It analyzes the trade-offs between these three designs in terms of privacy properties, auction properties, and fraud resistance. 3) It analyzes the effect of bid churn and auction timing on revenue and ad ranking using a trace of Bing search advertising auctions, and uses this analysis to argue for the feasibility of the solutions.

2. BACKGROUND

This section describes how current online advertising systems such as Google and Microsoft work, and then describes an abstract alternative advertising model patterned after existing proposals for *non-tracking advertising*. In the process, we establish terminology and define the basic components.

2.1 Current 2nd Price Ad Systems

In current ad systems [7,11,20], *advertisers* submit *ads* to a *broker*. Associated with each ad is a *bid*, one or more *keywords*, and optionally some targeting information like demographics (location, age, gender) or interests. When a *client* computer does a search or receives a web page with *adboxes* (space to place an ad), the broker identifies the ads that match the search terms or keywords associated with the web page, and runs an auction. The auction ranks the selected ads in order of highest expected revenue ($Bid \times ClickProbability$), and transmits some number of ads to the client. As already discussed, many factors are considered in estimating click probability. In this paper, we refer to all of these factors taken together as a *quality score* Q, where a higher value means higher expected click probability. Denoting bid as B, the ranking then is in order of the product ($B \times Q$).

When a user clicks on an ad, the ad ID is transmitted to the broker. The broker computes Cost per Click (CPC), that is, the price that the advertiser must pay, using a *second-price auction* [8]. In this approach, the price paid is pegged to the bid of the ad that is ranked immediately below the clicked ad. To give a simple example, suppose that advertiser A is willing to pay as much as \$5 for a click, and advertiser B is willing to pay \$10. In a second-price auction, A could go ahead and bid \$5 and B could bid \$10. B would win, but would only pay incrementally more than A's (2nd-price) bid, say \$5.01.

Second-price auctions allow bidders to bid the maximum that they are willing to pay, rather than frequently modify their bid in search of the value incrementally higher than the next lower bidder. Specifically, the CPC is computed as:

$$CPC = B_n \left(\frac{Q_n}{Q_c} \right)$$

where B_n and Q_n are the bid and quality score of the next lower ranked ad, and Q_c is the quality score of clicked ad. This CPC formula captures the minimum amount the advertiser would have had to bid to beat the next-ranked ad in a first-price auction. Note that it prevents the advertiser from paying more than it bid, even when the next-ranked in fact bid more.

2.2 Abstract Non-tracking Advertising

In this section, we describe an abstract non-tracking advertising system that captures key aspects of the three existing non-tracking advertising designs.

The principle components of the abstract non-tracking advertising system include those of today's tracking systems, the *broker*, the *client*, and the *advertiser*. A *user profile* is stored at the client (i.e. the user's computer, or a device trusted by the user: the distinction is not important for our purposes). Each ad is associated with targeting information. The user profile is defined as that information needed to determine how well an ad's targeting matches the user. To produce the user profile, the client monitors user behavior (i.e. the user's searching, browsing, purchases, and so on).

The privacy goals of the abstract non-tracking system are:

- Anonymity: the broker cannot associate any unit of learned information with any user PII (including network address), and
- Unlinkability: the broker cannot associate separate units of learned information with a single (anonymous) client. This prevents a broker from building up a user profile, and then associating it with a known user using externally gathered knowledge.

The broker is assumed to be honest-but-curious. We believe that this is close to reality (brokers like Google can generally be trusted to do what they claim they are doing). Nevertheless, we believe it is wise to avoid making it possible for brokers to obtain high-value information through hard-to-detect cheating, and our designs reflect this belief.

Figure 1 illustrates the basic architecture and message exchange of an abstract non-tracking advertising systems. The network layer address of all messages is anonymized, which we represent as an anonymizing proxy. Messages are encrypted to prevent viewing by the anonymizing proxy. The

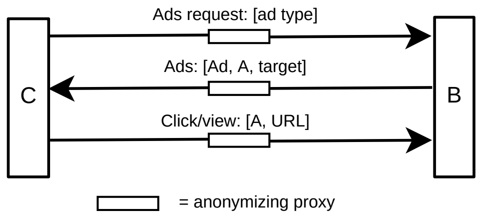

Figure 1: Abstract non-tracking advertising system. B is the broker, C is the client. Communications is through an anonymizing proxy. [x] denotes encryption of x.

client requests a set of ads of a given type (i.e. for a given product or service). The request must be generic enough that a substantial set of clients can have legitimately made the request (i.e. K-anonymity and L-diversity). A set of ads matching the type, each with identifier A and associated targeting information, are transmitted to the client and stored. When an adbox is presented to the client, for instance on a web page, the client selects among the stored ads those that best match the user profile, and puts them in the ad box for viewing by the user. The client reports the view and click of the ad A on a webpage with the given URL. There is nothing in the messages that allows the broker to link different messages as coming from the same user.

Our model has two necessary channels of communications between client and broker, ad delivery, and view and click reporting. Both channels present opportunities for the broker to learn information contained in user profiles, and current non-tracking advertising designs protect these channels. Any new information that must be conveyed for the purpose of the auction must also be protected.

3. AUCTION GOALS

The privacy goals for the auction component of a non-tracking advertising system are the same as described in Section 2: anonymity and unlinkability. This section describes the goals of the auction itself.

The primary goal of the auction in a non-tracking advertising system is to provide a second-price auction mechanism that achieves close-to-ideal ranking of ads (i.e., in order of $Bid \times ClickProbability$). For today's tracking advertising systems, leveraging the user profile is straightforward, since the broker itself accumulates and maintains this information. In a non-tracking system, the broker does not have user profile information, but does have other information that goes into the quality score Q. In other words, part of the information used to produce Q is in the broker, and part is in the client. Therefore, we define a *user score U* which directly reflects the effect of the user profile, when matched against an ad's targeting information, on click probability. We define a second quality score G, that reflects the remaining "global" information known to the broker.

Specifically, this results in an ideal ranking and CPC of:

$$Rank \Rightarrow B \times G \times U \qquad (1)$$

$$CPC = B_n \left(\frac{G_n \times U_n}{G_c \times U_c} \right) \qquad (2)$$

For example, U could be a positive real value greater or less than 1 that raises or lowers the click probability proportionally to its effect on the click probability defined by G. Section 7 briefly discusses how U may be computed.

In current tracking advertising systems, the click normally takes place almost immediately after the view, and so CPC is normally computed shortly after the ranking. As a result, the parameters that go into determining the ranking (B and Q) do not change much between ranking and CPC. In non-tracking advertising systems, as explained later, some time may pass between when B is set by the advertiser and when the ad is ranked, or between when an ad is ranked and when CPC is calculated. Therefore, we set the following goals with respect to ranking and CPC calculation:

- The B, G, and U used for CPC calculation are the same as the B, G, and U used for ranking. Note in particular that if they are not the same, then it is possible for instance for the CPC to be higher than the submitted bid of the clicked ad.
- The delay between ranking and CPC calculation is small enough that the churn in B, G, or U does not have a significant impact on rankings, CPC values, and broker revenue.

What exactly comprises user score U depends on the client profiler and can vary from system to system. We can, however, classify user information into three time frames. At the time frame of months or even years are user demographics like gender, location, language, age, salary, and so on. User interests can also last years (e.g. coin collecting), but more typically last weeks (a new car), days (a new pair of shoes), or minutes (a pizza). If we assume that matching ads to the content of a web page or search page increases click probability, then user score can change in seconds or less. For instance, a user might be interested in tennis and music, but the user score for tennis ads may increase while the user is looking at a tennis website, and vice versa for music ads.

We do not make any assumptions about the relative importance of B, G, or U. An ideal auction design however must allow for this flexibility.

Besides the basic goal of running an auction that leverages the user profile while prohibiting the broker from reconstructing it, there are a few additional related goals that are important:

- to maintain the privacy offered to the advertisers themselves. In particular, to prevent advertisers from learning each other's bids and budgets.
- to maintain the level of click-fraud defense in current tracking systems.
- to minimize the overhead of the auction.

As will become apparent in subsequent sections, our designs do not perfectly achieve all of these goals. Rather, our designs offer trade-offs between these goals.

Note finally that it is possible that the value of U may correlate with the probability that the user will buy the product or service being advertised assuming that the user has clicked. If this is the case, then the advertiser would want to express multiple bids as a function of U, since different U's would produce different revenues for the advertiser. We do not address this capability in this paper, but rather leave it for future work should it turn out to be important.

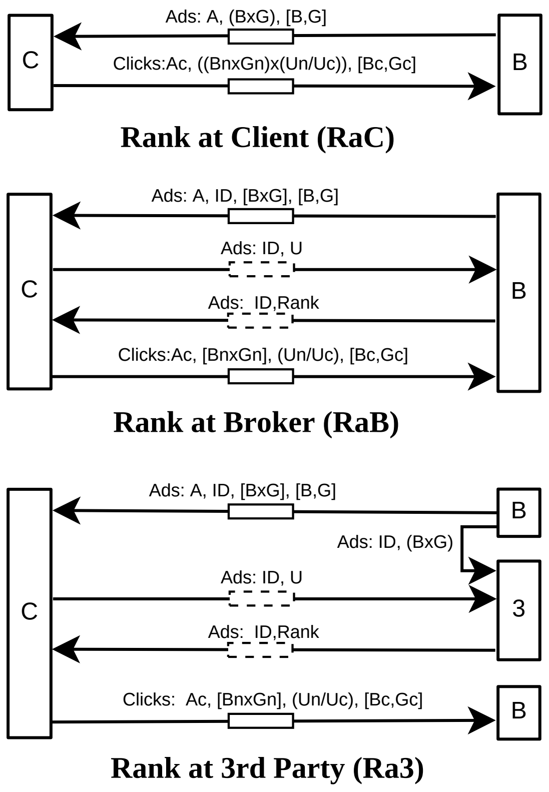

Figure 2: Three basic auction schemes. A is an ad ID unique to the ad, while ID is an Ad ID unique to the combination of ad and client. The subscripts 'c' and 'n' refer to the clicked ad and the next ranked ad respectively. [x] denotes encryption of x. Messages between client and broker pass through anonymizing proxies. Messages through solid-line proxies are encrypted. Messages though dotted-line proxies are in the clear.

4. DETAILED DESIGNS

To run the auction specified in Section 3, the system doing the ranking must have access to the bid B, the broker quality score G, and the user score U. This means that either 1) B and G are sent to the client, 2) U is sent to the broker, or 3) B, G, and U are all sent to a 3rd party (Figure 2). These basic approaches are explored in the following sections.

4.1 Rank-at-Client (RaC)

In this approach, the following information is transmitted with the ad to the client along with everything else required by the advertising system (e.g. targeting information, not shown):

A: The ad ID.

$(B \times G)$: A single value which is the product of $(B \times G)$.

$[B, G]$: The individual values B and G, encrypted with a symmetric key known only to the broker.

When a collection of ads arrive at the client, it ranks all ads using $(B \times G \times U)$. Note that in this case U is current, while $(B \times G)$ is more-or-less older. If the user clicks on an ad, then the client computes the following values and transmits them to the broker:

A_c: The ad ID of the clicked ad.

$((B_n \times G_n) \times (U_n/U_c))$: A single value which is the $(B_n \times G_n)$ product of the next-ranked ad times the ratio (U_n/U_c) of the user score of the next-ranked ad U_n and the user score of the clicked ad U_c.

$[B_c,G_c]$: The encrypted B and G for the clicked ad as received earlier from the broker.

Upon reception of this message, the broker decrypts $[B_c,G_c]$. It uses G_c and $((B_n \times G_n) \times (U_n/U_c))$ to compute the CPC as shown in Equation 2. The broker also compares the resulting CPC with the decrypted B_c. If $CPC > B_c$, then the broker knows that the client is engaged in click fraud, and the broker can ignore the message. If $CPC \leq B_c$, then the broker can accept the message, although this doesn't mean that the client is not engaged in click-fraud. Other mechanisms, such as statistical analysis, must be used to detect it as is done today.

Variation: It may not be necessary to transmit the encrypted values $[B_c,G_c]$. This is because B_c and G_c can be looked-up using the ad ID A_c. The danger here is that the looked-up values may be different from the B_c and G_c values used to rank the ads. How different depends on the level of churn in B and G values, which we found to be minimal in the Bing auction trace (Section 6). Therefore, it may well suffice to use looked-up values rather than values stored along with the ad at the client. Note that this variation applies to RaB and Ra3 as well.

4.2 Rank-at-Broker (RaB)

One concern with RaC is that the value (BxG) exposes information about the advertiser (see Section 5.1.3). This can be avoided if the ranking is done at the broker. The RaB scheme presented here is similar to the approach proposed by Privad in [12]. We present it here for completeness.

Along with the ad, the broker transmits the following to the client:

A: The ad ID.

ID: An identifier unique to this specific delivery of this ad (among all other deliveries). In other words, the same ad delivered to other clients would have a different values of ID.

$[B \times G]$: A single value which is the product of $(B \times G)$, encrypted with a symmetric key known only to the broker.

$[B, G]$: The values B and G, encrypted with a symmetric key known only to the broker.

The client computes a user score U for each ad (in the absense of knowledge of what web page the ad may be shown on). In order to obscure the user profile, the client assigns a random value for U for those ads for which the client has a very low user score (for instance because the demographic doesn't match that of the user).

Clients transmit the ID,U tuples to the broker via a proxy. These messages are not encrypted. The proxy also remembers which IDs were received from which clients.

For each received ID, the broker looks up the current values of B and G. It then uses B, G, and U to rank each received ad among a large number of recently received ads (say, those received over the last hour), and associates a rank number $Rank$ to each ad. Closely ranked ads may have the same ranking number. The broker transmits the ID,$Rank$ tuple back to the proxy.

The proxy looks up which client is associated with each ID, and forwards the message on to the client. The client disregards the ads related to the low user scores, and uses the remaining ranking for selecting ads to put in ad boxes.

When a user clicks on an ad, it transmits the following information to the broker:

A_c: The ad ID of the clicked ad.

$[B_n \times G_n]$: The encrypted $(B \times G)$ for the next-ranked ad received earlier.

(U_n/U_c): A single value which is the ratio of the user score of the next-ranked ad U_n and the user score of the clicked ad U_c.

$[B_c,G_c]$: The encrypted B and G for the clicked ad received earlier.

Upon reception of this message, the broker decrypts $[B_n \times G_n]$ and $[B_c,G_c]$. It uses G_c, $(B_n \times G_n)$, and (U_n/U_c)) to compute the CPC as shown in Equation 2. As with RaC, the broker also compares the resulting CPC with the decrypted B_c for click fraud.

The proxy prevents the broker from learning the identity of the client whose ads are being ranked. The per-client-per-ad unique ID prevents the proxy, which does know the client identity (network address), from learning which ad, and therefore what targeting information, is being referred to.

Even with the noise added to low user scores, we are concerned that the non-noise user scores can be interpreted at the broker as a kind of fingerprint over the set of ads (i.e., ads targeted to men should have uniformly higher scores for men, and lower scores for women). In this way, the broker could potentially tease out the profile of users.

4.3 Rank-at-3rd-Party (Ra3)

This approach is similar to RaB, but prevents the fingerprinting mentioned above. The main difference between Ra3 and RaB is that in Ra3, the broker additionally sends the unique ad IDs and $(B \times G)$ products to a 3rd party system which is trusted not to collude with the broker. This information must be delayed long enough that the 3rd party system cannot use a timing attack to correlate the values associated with a single user. This 3rd party also receives the user scores from the clients, and based on this information, ranks ads in the same way the broker does in the RaB approach. Since, unlike the broker, it does not know which ads were transmitted to the same client, it cannot fingerprint the clients.

4.4 Homomorphic Encryption Variant (RaC, RaB, and Ra3)

A variation on all three auction designs is to use homomorphic encryption (e.g., ElGamal [9]), which allows for multiplication operations on encrypted data. This may be used to defend against certain attacks by the broker as described in Sections 5.1.1 and 5.3.3. When a user clicks on an ad, the client encrypts (U_n/U_c) with the broker's public key. In the case of RaC, it also encrypts $(B_n \times G_n)$ with the broker's public key. In the case of RaB and Ra3, the broker provides the encrypted $[B_n \times G_n]$, but using its public key instead of a symmetric key. For all three schemes, the broker provides $[1/G_c]$, again encrypted with the broker's public key. Using homomorphic property of the encryption, the client is able to calculate:

$$[B_n \times G_n] \times \left[\frac{U_n}{U_c}\right] \times [1/G_c] = \left[B_n\left(\frac{G_n \times U_n}{G_c \times U_c}\right)\right]$$

and transmit the resulting value in the click report. Upon receiving a click report, broker decrypts the value to obtain the CPC. Although homomorphic encryption is relatively expensive, there is no need to do the operation in real-time. Rather, the client can do the operation when it has spare CPU cycles before transmitting, and the broker can likewise run the operations later on as batch processing.

5. AUCTION ANALYSIS

This section analyzes the three types of auctions in terms of privacy, auction quality, and attacks on the auction systems.

5.1 Privacy Properties

In this section, we look at the information that is conveyed for the sake of the auction between honest-but-curious players, and determine whether it constitutes a privacy threat. In Section 5.3 we relax this assumption.

5.1.1 Broker analyzes (U_n/U_c) (RaB and Ra3)

In order to exploit this value to gather more information about the user profile, the broker would have to first tease apart the values of U_n and U_c, then use the value combined with the ad targeting to reverse engineer the user profile, and then use the user profile knowledge to link together multiple reports. The first step may be made difficult by making (U_n/U_c) relatively coarse-grained, thus making it harder to uniquely factor out its components. The second step is made difficult simply by the sheer number of clients that are likely to have similar user scores. Thus, we conclude that exposure of (U_n/U_c) does not constitute a serious threat.

5.1.2 Broker analyzes $((B_n \times G_n) \times (U_n/U_c))$ (RaC)

This value is more difficult to reverse engineer than (U_n/U_c), and is therefore also not a threat.

5.1.3 Client analyzes $(B \times G)$ (RaC)

An advertiser can use this value to determine the broker quality score G assigned by the broker to its own ads. This can be done by the advertiser simply creating a client that receives its own ads, and using the known value of B to factor out G. Whether this is a problem needs to be decided by the broker, though we point out that today Google reveals a coarse-grained quality score to its advertising customers.

The product $(B \times G)$ to the client also reveals the overall ranking of an ad to anyone running a client, including the advertiser's competitors. From this, they can also roughly estimate the advertiser's bids. It is not clear that this is a problem, for two reasons. First, in today's advertising systems, an advertiser can see how its competitors rank relative to itself simply by observing how ads are displayed. RaC makes it easier and cheaper to obtain this ranking information, but does not fundamentally change an advertiser's ability to do so. Second, historically in traditional advertising (print, TV, radio), advertisers can monitor how much advertising their competitors do, and can generally know the cost of that advertising. While certainly all things being equal advertisers would like to keep this information secret, historically the inability to do so has not, for the most part, prevented companies from advertising.

If exposing the product $(B \times G)$ to the client is an acceptable privacy loss, then RaC should be the preferred auction method for its overall simplicity and lower overhead. If it is not acceptable, then RaB and Ra3, which both avoid exposing this information, may be preferred.

5.2 Auction Properties

In this section, we discuss the various shortcomings of each of the approaches with respect to the auction properties, especially ranking results and revenue.

5.2.1 System delays

There are several potential delays in the non-tracking advertising systems that can change both the rankings and the computed CPC. With RaC, there is a delay between when the ad was transmitted and adbox time when the ranking takes place. With RaB and Ra3, there is a delay between when the ranking occurs and adbox time when the ranking is actually used. In either case, the bid B or the broker quality score G used for ranking may no longer be correct, and an out-of-date ranking takes place.

During the design of the auction approaches, these delays were a major concern. As it turns out, at least for the auction data from Bing search advertising auctions (Section 6), the delays have only a minor impact on both broker revenue and advertiser costs, even when the delay is several hours or a day. Nevertheless, this may not be the case for other systems or future systems, and so it remains important that these delays are engineered to be minimal. This could be done, for instance, by having clients frequently request small numbers of new ads.

5.2.2 Client selection

A problem encountered by non-tracking advertising is that the broker does not know which clients are the best clients to send an ad to. For instance, suppose that some number of clients M have requests ads for watches. The broker does not know which clients may be interested in cheap watches, and which in expensive. The advertiser, however, might not have enough budget to pay for all the clicks that would result if all watch ads are sent to all interested users.

Lets assume that the broker knows the clicks per delivered-ad rate. From this, it can determine the number of clients N that should receive the ad without exhausting the advertisers budget. If it randomly chooses N clients among the interested clients, then it will not be sending all ads to the most interested clients.

One way to solve this problem is for the broker to go ahead and send the ad to all interested clients, but to also send a parameter giving the minimum user score that a client must have in order to show the ad. This way, only the best matching clients will show the ad. The broker may be able to establish the expected click per delivered-ad rate for various user scores, and therefore predict the setting of the user score based on the number of clients. If the broker predicts too high, then it can lower the minimum user score and send this to clients, thus causing more clients to show the ad.

5.2.3 Auction Scope

An important aspect of the auction is the scope of the auction, by which we mean the set of ads that compete in any given auction. As a general rule, the more ads that compete, the higher the CPC. This is simply because the more ads there are, the more probabilistically likely the next-ranked ad will have a $(B \times G \times U)$ closer to that of the clicked ad. On the other hand, the larger the auction scope, the less fair it is in the sense that very different types of ads must compete. A local pizza store may not wish to compete with Mercedes for ad boxes.

The auction scope for search or contextual systems like Bing and Google is the set of ads whose keywords match that of the search or web page. Today's ad exchanges, where advertisers bid in real time, typically for adboxes on premium publishers, have a potentially much broader scope because any advertiser can bid. The auction scope in a non-tracking advertising system is tunable. It may be all ads in a client, or all ads within an ad type (i.e. an interest). What's more, interests may be hierarchical (sports/tennis/clothing/shoes), and may be more general or more specific, thus allowing for substantial flexibility in auction scope.

5.3 Attacks

In this section, we relax our assumption of honest-but-curious players, and consider a number of malicious attacks and defenses.

5.3.1 Client click fraud

The client can commit a form of click fraud by lying about the value of $((B_n \times G_n) \times (U_n/U_c))$ (RaC) or (U_n/U_c) (RaB or Ra3). By inflating or deflating these values, the client can cause advertisers to pay more or less, and cause publishers to earn more or less. At a high level, this is very similar to normal forms of click fraud that occur today, and in this sense our auctions do not allow fundamentally new forms of click fraud. Privad describes how to defend against click-fraud even with anonymizing brokers [12]. The same method may be used here. The basic idea is that the proxy tags reports with a per-report unique identifier. If the broker suspects click fraud, it informs the proxy of the report ID of the suspicious report. If a given client is suspected more times than some threshold, its reports can be tagged by the proxy as coming from a suspected client. In some cases the broker may suspect click fraud simply because the 2nd price is impossibly high (i.e. higher than the 1st price bid). In most cases, however, the broker may suspect click fraud through statistical analysis of the reports for given advertisers or publishers.

5.3.2 Proxy fingerprints client user scores and resulting ranking (RaB and Ra3)

It is difficult but conceivable in RaB and Ra3 that the proxy could determine user profiles through observation of the client user scores and rankings. For instance, the proxy could establish a number of fake clients that pretend to have various profile attributes, and establish fingerprints of the resulting user scores and rankings. One way to do this might be to determine $(B \times G)$ given user scores and corresponding ad ranks, and use these values as the fingerprint. The proxy could then compare these fingerprints with the corresponding fingerprints of real clients. It could be that the signal-to-noise ratio is high enough to successfully pull off this attack. One way to prevent this would be to encrypt user scores and rankings. The user scores could be encrypted using the brokers (RaB) or 3rd-party's (Ra3) public key, and the rankings could be encrypted using symmetric keys created by the clients and conveyed securely to the broker or 3rd-party. These symmetric keys would be frequently modified to prevent the broker or 3rd-party from linking user scores with the same client, and possibly launching a similar fingerprint attack.

5.3.3 Broker manipulates $[B_n \times G_n]$ (RaB, Ra3)

A malicious broker could launch an attack on a non-tracking advertising system to identify clients by inserting unique IDs into the encrypted fields $[B, G]$ or $[B \times G]$. Once a client is identified in this way, unlinkability is lost, and the broker can build up client profiles. The broker can then potentially identify the client through external means. There is some cost to this approach, as the broker must "waste" ads to do the tracking[2]. The homomorphic encryption variant described in Section 4.4 defends against this attack. Because the client multiplies the received encrypted fields with other fields, the values generated by the broker are obscured.

5.4 Discussion

RaB, which was proposed in [12], is a weak system because it opens up a fingerprinting attack at the broker. Ra3 solves this problem, though at the expense of requiring yet another administratively distinct and non-colluding system. Nevertheless, we consider it to be better than RaB.

If exposing the $(B \times G)$ product to the client is not a problem for the broker and advertisers, then RaC is better than Ra3 because it is simpler, incurs less overhead and latency, and does not require the 3rd party. In addition, it has no issues here with respect to user score churn, because ranking takes place at ad view time. If exposing $(B \times G)$ is a problem, however, then Ra3 appears to be a reasonable approach.

6. EFFECT OF CHURN

Section 5.2 describes how various delays in all three auction systems may distort rankings and CPC computation. How detrimental this delay is depends on how much churn there is. Churn may affect rank, CPC value, and ultimately revenue. RaC is affected by churn in B and G, while RaB and Ra3 are additionally affected by churn in U. In this section, we use trace data from Microsoft's Bing advertising platform to study in depth the effect of auction delays on B and G from both the advertiser and broker perspective. We find that, while churn exists, it has only a negligible impact on broker revenue and advertiser costs.

6.1 What Causes Churn?

$B \times G$ for an ad changes when either B or G changes. B can change in one of three ways: first, the advertiser can manually update the bid; second, the ad network can automatically update the bid (as directed by the advertiser); third, a 3rd-party may update the bid on the advertiser's behalf. Each of these has different churn characteristics:

Advertiser: Manual updates, we believe, cause very little churn since they are reactive over a long feedback cycle. Advertisers receive updated campaign information (i.e. how many clicks, actual amount charged, budget left) at fairly coarse intervals (few times a day). This limits the number of informed changes to their campaigns.

Ad Network: The advertiser can invoke functionality provided by the ad network to optimize his bidding strategy. For example, the ad network may allow the advertiser to set a preferred rank (e.g. position 4), and the ad network automatically lowers or raises the bid to satisfy the request based on the market. Other examples may include automatically modifying bids to meet a target number of impressions per day (while still being charged only for clicks), or modifying bids based on time of day etc. Some of this functionality (e.g. modify bids based on time-of-day) can be implemented in the client and would therefore not result in any added churn. Other functionality (e.g. preferred rank) tends to be implemented today as a periodic update (once every few hours).

3rd Party: Search Engine Optimization (SEO) companies optimize their client's bidding strategy in real-time [6] e.g. based on trending terms, real-time click-through rates, etc. This could potentially result in high bid churn, however, due to the premium nature of these SEO services, only a small number of ads would be affected.

Aside from changes in B, G can also change. Recall G in our model is a function of what the broker knows: G is computed based on the ad (past CTR, landing page quality, etc.). G is largely a property of the ad itself, which we don't expect to change quickly or dramatically. In any event, our Bing auction trace unfortunately does not allow us to validate our assumption since it does not isolate user-derived components of G from other components.

6.2 How Does Churn Affect Auctions?

Today auctions take place at the time when an ad is displayed to the user; ranking and CPC calculations can immediately reflect any changes in B or G. Privacy preserving auctions described in Section 4 are limited in terms of how fast new B and G information can be incorporated. Since G does not rapidly change over time or can be engineered to remain relatively stable (e.g., using U to reflect short-term changes in click probability), the main source of churn is the changes in B. To understand the effects of churn in B values, we simulate auctions that use stale B information for ranking and CPC computation, and then compare the resulting ranking and CPC computation with auctions that use up-to-date B information.

6.3 Dataset

For our trace driven simulations, we sampled around 2TB of log data from Bing's auction engine spanning a 48 hour period starting September 1, 2010. The data covers over 150M auctions for over 18M unique ads shown to North American Bing search users across all search topics. The trace record for an auction lists all the ads that participated in it (whether the ad was ultimately shown or not), the bids corresponding to each ad, the corresponding quality scores,

[2]One might argue that the same attack can be launched simply be creating unique Ad IDs transmitted in the clear. However, this attack can at least be detected by third parties, for instance running honey-farms of clients.

and which if any of the ads were ultimately clicked by the user.

6.4 Methodology

We re-compute auction rankings and the CPC for each auction in our dataset using stale bid information; we vary staleness from 1 minute to 2 days.

Auction rankings are re-computed using bid and quality data from the trace. Since our trace does not show when the advertiser updated the bid, we infer the time based on multiple auctions that a given ad participates in. If the bid for the ad is the same for two consecutive auctions, we infer that the bid did not change during that interval. If the bid is different, we infer that the bid changed sometime between the two auctions; we use the mid-point as the time of change. To simulate an auction at time T with stale information from d minutes ago, we simply use the bids current as of time $T - d$ in our trace. The quality score in the trace is based on user features (e.g. search query), which correspond to U in private auctions; since the client always has the current value of U we use the same quality score for simulated auctions as in the trace.

CPCs are re-computed based on the re-computed auction rankings (using the second-price formula of Equation 2). In other words, for an adbox at time T in the trace, we compute the ranking based on bid values recorded at time $T - d$ and populate the adbox using resulting ranking. If the user clicked on an ad in this adbox, the bid of the next lower ranked ad B_n that we use in the CPC computation is the stale B_n taken at time $T - d$.

One limitation we face is that we cannot predict the change in user behavior when auction rankings change. Consider, for example, two ads A_1 and A_2 where in the trace they are ranked 1 and 2, while in the simulated stale auction they are ranked 2 and 1 respectively. If the user clicked A_2 in the trace, what might we expect the user to click in our simulation? One option is to model the user as clicking the same ad he clicked in the trace; thus in this case the user clicks A_2 in the simulation. Another option is to model the user as clicking the same position he did in the trace; in this case the user clicks position 2 (A_1 in the simulation). In reality, the user model is neither of these two extremes — it is well-known that both ad content and rank effect click-through rates (CTR) [10]. To account for this, we simulate 5 user models: 1) same position, 2) 75% same position and 25% same ad, 3) 50%-50%, 4) 25% same position and 75% same ad, and 5) same ad. Thus we establish an envelope of possible user behavior to get a sense of the upper- and lower-bounds of our simulation results. Note that always clicking on the same ad is a strictly conservative estimate. This is because an ad that was clicked in the trace but is not shown to the user in our simulation (due to being ranked too low) would not get clicked; at the same time, under the same-ad model, an ad that was not shown in the trace (due to being ranked too low) and was therefore not clicked would have no chance of getting clicked even if it were to be shown to the user in the simulation. This asymmetry biases the simulation towards fewer clicks (and therefore lower revenues). The only user model immune to this limitation is the same-position model.

A second limitation we face is that we cannot predict how advertisers would change their bidding strategy in response to auctions being based on stale information. Enterprising advertisers or SEOs, may for instance, attempt to predict what bid they might want to make 1hr hence, and enter it into the system well in advance. Advance bidding would reduce the effective staleness of information. For our purposes, we assume the bidding strategy does not change.

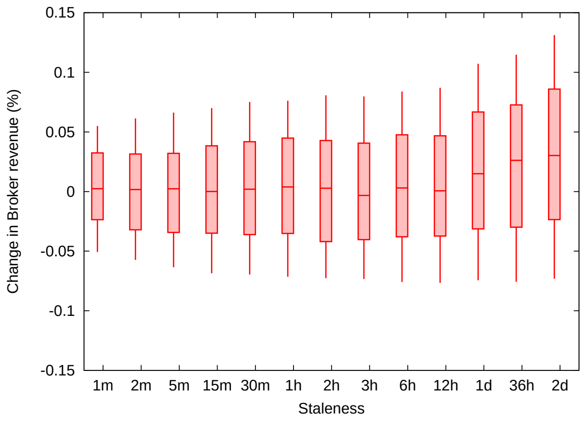

Figure 3: Change in broker revenue

6.5 Simulation Results

Overall our simulations show that there is no appreciable change in broker revenue for using stale bid information; even in the most conservative cases, the revenue is within ±0.1% of today. For advertisers, while stale bids affect their auction rankings, they do so in a balanced manner with cases of higher-than-today rank canceling out cases of lower-than-today rank resulting in zero net change.

Figure 3 plots the change in broker revenue compared to today as a function of the staleness of information used and the user model. The x-axis varies the stateless of bids from 1 minute to 2 days. The box-and-whisker plot varies the user model with the top whisker showing the outcome where the user clicks the same position, and the bottom whisker showing when the user clicks the same ad; the top edge of the box shows 75% same position and 25% same ad, and vice versa for the bottom edge of the box; the line in the middle shows the 50%-50% case.

The first observation we make from Figure 3 is that under a 50-50 user model, change in revenue is practically 0% even with bid information as stale as up to 12 hours. Under the 75-25 and 25-75 models, the change is almost always between ±0.05%, and only in the extreme cases 100-0 or 0-100 does it pass ±0.1%. More importantly, the change increases very gradually. This is good news since it means a private advertising system would not have a hard delay deadline beyond which there would be disproportionate change in revenue. Instead the system can strive to do the best it can, and reduce revenue change proportionally. The extremely gradual rate of change also means that system design trade-offs can be biased towards scalability and other engineering goals without much concern to revenue since it changes very little in the first place.

At first blush the effect of the “same-ad” user-model appears to be to reduce the revenue, but this is deceptive. As mentioned earlier, the more the user clicks on the same-ad (going from 0% to 100% from the top whisker to bottom), the more biased the simulation is towards fewer clicks and

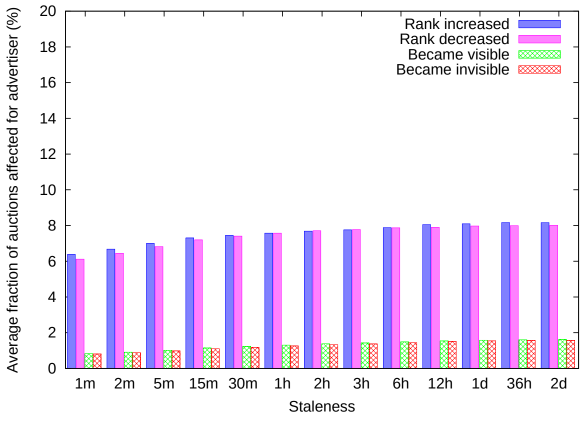

Figure 4: Fraction of auctions with modified rankings

therefore less revenue. Recall that only the top-whisker is unaffected by this simulation bias.

The second observation we make from Figure 3 is the slight upwards trend of the top-whisker signifying higher revenues as more stale information is used. This suggests a consistent trend of advertisers (as a whole) reducing their bids over time. We don't know the cause of this trend.

Next we turn to the advertiser perspective. We compute for each ad the fraction of auctions where the user-visible simulated ranking increased or decreased compared to the trace, and whether the ad became visible or invisible due to being ranked high-enough or too-low as compared to the trace. Figure 4 plots the average of these numbers across all ads as a function of the staleness of bid information used.

We first observe that both increased ranks and decreased ranks are roughly equal, and so average to nearly zero. The same is true for ads becoming visible or invisible. While this is consistent with the revenue change in Figure 3 averaging out to zero, we note that there are other ways the revenue could average out to zero while being unfair to advertisers. For instance, fewer increased ranks could have been compensated by more cases where the ad became visible thus still resulting in zero revenue change while being unfair to the advertiser; luckily, this is not the case.

We observe next that there is a very small impact of staleness on change in ranks; it begins with around 12% of auctions for 1 minute stale data, and quickly converges to around 16%. The reason this number is high is because of the cascade effect — if a single ad jumps from a low rank to a high rank, it causes all the ads in between to register a "change" in rank; thus a single change in bid can affect up to ten ads. The impact, however, is very little; the ad jumping from low to high might register a change of 10 ranks, however, the other 10 ads would register a change of only 1 rank each (not captured in the graph). Overall we found a median *net change* of 1 rank for every 820 auctions the ad participates in.

To summarize, based on extensive simulations across varying degrees of staleness and different user-models, there is little impact on broker revenue as compared to today, and little impact on advertiser fairness as compared to today.

7. COMPUTING USER SCORE

So far, we have assumed the existence of a user score U that, when multiplied with the quality score G produces the expected click probability at the client for a given ad. Because clicks are relatively rare, it may be difficult to estimate U at the client based purely on the click history of the client. Therefore, we require that the broker anonymously and unlinkably gathers detailed click statistics from clients in order to improve click probability estimates at individual clients. In what follows, we sketch out an approach.

There are a number of measurable attributes $X = \{x_1, x_2, ...x_L\}$ at the client that may help in prediction of click probability. For instance, the level of interest (high or low) in the ad's product or service, the quality of the match between the targeting and the user, the context of the webpage, as well as the user's historic CTR. The idea is that each client reports this information anonymously to the broker for each ad that it views and clicks. These reports contain: $\{$Ad-ID, X, click$\}$, where X is the values of the attributes, and 'click' indicates whether or not the ad was clicked. Given this information from many clients, the broker can determine the effect of the attributes on click probability, and convey this information to the client as a function f of the attributes such that $U = f(X)$, along with the ad. This allows the client to compute U by measuring the attributes and plugging them into the provided function. As mentioned, U in RaC can be computed at viewing time with the latest set of attributes without churn issues since that's when the ranking takes place. The function f for a new ad can be initially set to that of similar existing ads until enough data for the new ad is gathered. The details of this are left as future work.

One concern is that the set of attribute values X is unique for a given user. Several factors can mitigate this concern. First, the attributes may be fairly coarse-grained, thus broadening the set of users to which they apply. Second, some of the attributes may be hard to correlate using external knowledge, such as the user's CTR. Third, attributes like level of interest change from interest to interest, and even within an interest over time, and therefore are hard to link to the same user. Fourth, some attributes are not specific to the user, for instance webpage context. Finally, the only information beyond the attribute values that is leaked is the ad viewed. In particular, the user's click-stream is not exposed. We believe that it is reasonable to establish public policies that determine the nature of the attributes in such a way that meaningful privacy is preserved.

8. RELATED WORK

There is a substantial body of work on cryptographic protocols for privacy preserving auctions. Depending on the underlying security model these proposals can be classified into the following three categories. In the first category, there are protocols that rely on computation that is distributed among auctioneers who jointly determine the outcome of an auction using threshold multi-party computation (e.g., [14, 15, 22]). The second category of protocols introduces a semi-trusted third party, aka an "auction issuer" or "auction authority", in addition to the auctioneer, and uses asymmetric multi-party computation technique, such as Yao's garbled circuit (e.g., [1, 2, 5, 19, 21]). Finally, protocols in the third category allow bidders to cooperatively compute the auction outcome without relying on any trusted third party (e.g., [3,4]). The primary goal of all these proposals, and many others not cited, is to keep bids and selling price secret from the auctioneer and other auction participants. The problem that we address in this paper is different. This paper is primarily concerned with protecting the user, not the bidder (i.e. advertiser). Indeed the user does not exist in the prior work. In any event, the high computational and communication complexity imposed by aforementioned secure auction protocols make them impractical for our problem.

9. SUMMARY AND FUTURE DIRECTIONS

This paper addresses the challenge of designing an online advertising auction for a non-tracking advertising system that leverages the user profile information while keeping the user profile private. We broadly explore the design space, proposing three types of auctions, and analyzing their properties with respect to privacy, auction quality and vulnerability to attack. Overall, we find that two of the systems, Rank-at-Client (RaC) and Rank-at-3rd-Party (Ra3), are very acceptable designs. RaC is simpler and more efficient, but has the drawback that information about ad quality and bid is leaked. On the other hand, this is information that can today be determined by placing ads and monitoring the resulting ranking. Finally, noting that our auction designs suffer delays that cause out-of-date bid information to be used in rankings, we use Bing advertising system auction trace to determine the effect of these delays. We find the effect to be very minimal, and so conclude that our auction designs are viable.

As future work, we plan to implement the auction system to operate with a medium-scale deployment of Privad planned for next year (several 10's of thousands of users). Along with this, we plan to design and deploy mechanisms to compute U, and measure their effectiveness. We also plan to do a measurement study of ads served by Bing to determine to what extent advertisers can reverse-engineer each other's bids in today's systems. This will quantify how much advertiser privacy loss is incurred by the Rank-at-Client scheme. Each of the non-tracking advertising schemes so far proposed makes the tacit assumption that there is only a single broker, and a single profiler operating at each client. We are interested in exploring what happens if there are multiple brokers with competing profilers in each client. In particular, the profilers should be able to dynamically compete for ad boxes in real time, thus adding a new element to the auction that is not unlike the way ad exchanges operate today. What's more, these clients may also need to compete with existing tracking advertising systems in real-time auctions run by existing ad exchanges. We plan on designing and testing mechanisms that allow this.

10. REFERENCES

[1] M. Abe and K. Suzuki. M+ 1-st Price Auction Using Homomorphic Encryption. In *Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptosystems*, pages 115–124. Springer-Verlag, 2002.

[2] O. Baudron and J. Stern. Non-interactive Private Auctions. In *Proceedings of the 5th International Conference on Financial Cryptography*, pages 364–378. Springer-Verlag, 2002.

[3] F. Brandt. Fully private auctions in a constant number of rounds. In *Proceedings of the 7th international conference on Financial cryptography*, pages 223–238. Springer-Verlag, 2003.

[4] F. Brandt and T. Sandholm. Efficient privacy-preserving protocols for multi-unit auctions. In *Proceedings of the 9th international conference on Financial cryptography and Data Security*, pages 298–312. Springer-Verlag, 2005.

[5] C. Cachin. Efficient private bidding and auctions with an oblivious third party. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 120–127. ACM, 1999.

[6] S. Clifford. Instant Ads Set the Pace on the Web. *The New York Times*, 2010. `http://tinyurl.com/yl8dt29`.

[7] DoubleClick. DART for Advertisers. `http://www.doubleclick.com/products/dfa/index.aspx`, 2009.

[8] B. Edelman, M. Benjamin, and M. Schwarz. Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords. *American Economic Review*, 97(1):242–259, Mar. 2007.

[9] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Information Theory, IEEE Transactions on*, 31(4):469–472, 2002.

[10] J. Feng, H. Bhargava, and D. Pennock. Implementing sponsored search in web search engines: Computational evaluation of alternative mechanisms. *INFORMS Journal on Computing*, 19(1):137, 2007.

[11] Google, Inc. AdWords: Advertise Your Business on Google. `http://adwords.google.com/`.

[12] S. Guha, B. Cheng, and P. Francis. Privad: Practical Privacy in Online Advertising. In *Proceedings of the 8th Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, Mar 2011.

[13] S. Guha, A. Reznichenko, K. Tang, H. Haddadi, and P. Francis. Serving Ads from localhost for Performance, Privacy, and Profit. In *Proceedings of HotNets '09*.

[14] H. Kikuchi. (M+ 1) st-Price Auction Protocol. In *Proceedings of the 5th International Conference on Financial Cryptography*, pages 351–363. Springer-Verlag, 2002.

[15] H. Kikuchi, S. Hotta, K. Abe, and S. Nakanishi. Distributed auction servers resolving winner and winning bid without revealing privacy of bids. In *Parallel and Distributed Systems: Workshops, Seventh International Conference on, 2000*, pages 307–312. IEEE, 2000.

[16] B. Krishnamurthy and C. Wills. On the Leakage of Personally Identifiable Information Via Online Social Networks. In *Proceedings of WOSN '09*.

[17] B. Krishnamurthy and C. Wills. Privacy diffusion on the web: A longitudinal perspective. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*, Madrid, Spain, 2009.

[18] D. Levin, B. Bhattacharjee, J. R. Douceur, J. R. Lorch, J. Mickens, and T. Moscibroda. Nurikabe: Private yet Accountable Targeted Advertising. Under submission. Contact `johndo@microsoft.com` for copy, 2009.

[19] H. Lipmaa, N. Asokan, and V. Niemi. Secure Vickrey auctions without threshold trust. In *Proceedings of the 6th international conference on Financial cryptography*, pages 87–101. Springer-Verlag, 2002.

[20] Microsoft, Inc. Start advertising on Yahoo! Search and Bing. `https://adcenter.microsoft.com/`.

[21] M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and mechanism design. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 129–139. ACM, 1999.

[22] K. Sako. An Auction Protocol Which Hides Bids of Losers. In *Proceedings of the Third International Workshop on Practice and Theory in Public Key Cryptography*, pages 422–432. Springer-Verlag, 2000.

[23] Stanford. Do Not Track Universal Web Tracking Opt-Out. `donottrack.us`.

[24] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas. Adnostic: Privacy Preserving Targeted Advertising. In *Proceedings of NDSS '10*.