PiCoDa: Privacy-preserving Smart Coupon Delivery Architecture

Kurt Partridge[1*], Manas A. Pathak[2*], Ersin Uzun[3], Cong Wang[4*]

[1]Google Inc.
[2]Adchemy Inc
[3]Palo Alto Research Center
[4]Illinois Institute of Technology

Abstract

In this paper, we propose a new privacy-preserving smart coupon delivery system called PiCoDa. Like prior work on private behavioral targeted advertising, PiCoDa protects user data by performing targeting on the end-user's device. However, PiCoDa makes two more guarantees: it verifies a user's eligibility for a coupon, and it protects the vendor's privacy by not revealing the targeting strategy.

To accommodate the constraints of different targeting strategies, PiCoDa provides two targeting protocols that tradeoff user privacy and vendor privacy in different ways. We show how both designs meet requirements for user privacy, vendor protection, and robustness. In addition, we present simulation results of the protocols using realistic parameters to further validate the efficiency and effectiveness of PiCoDa.

1 Introduction

In recent years, online advertising has come to rely more heavily on behavioral targeting.Behavioral targeting allows vendors to provide more relevant messages by using indicators of interest from historical data about a user. From the user's perspective, better targeting is beneficial as it leads to more personalized service and less exposure to information that is not of interest to them. Particularly interesting is how behavioral targeting incorporate not just web data, but physical contextual data like location, time of day, and proximity to other individuals [9].

However, despite the incentives to both vendors and users, the current practice of behavioral targeting raises great privacy concerns among users [15]. In order to target accurately, vendors need to know adequate information about users, such as their demographics, geographic locations, purchase behaviors, browser and internet search histories. However, collecting such information is usually in conflict with user privacy. Enabling accurate behavioral targeting without compromising user privacy is a challenging problem.

In this paper, we propose a new privacy-preserving smart coupon delivery architecture called PiCoDa. Instead of doing the behavioral matching for coupon requirements on the vendor side, we propose to perform it on the user device. This allows a vendor to deliver targeted coupons while users keep full control of their behavioral data, which never leaves the user's device. Shifting the behavioral targeting computations to the client device is not a new idea and has been recently applied to the context of personalized search [16] and online advertising [12, 7, 6, 3]. However, targeted coupon delivery poses additional challenges beyond those demanded by targeted advertising. Shifting the computations for behavioral matching to the user's device alone is insufficient to protect both users and vendors. First, coupons must be delivered only to the

*Work was done when all authors were affiliated with PARC

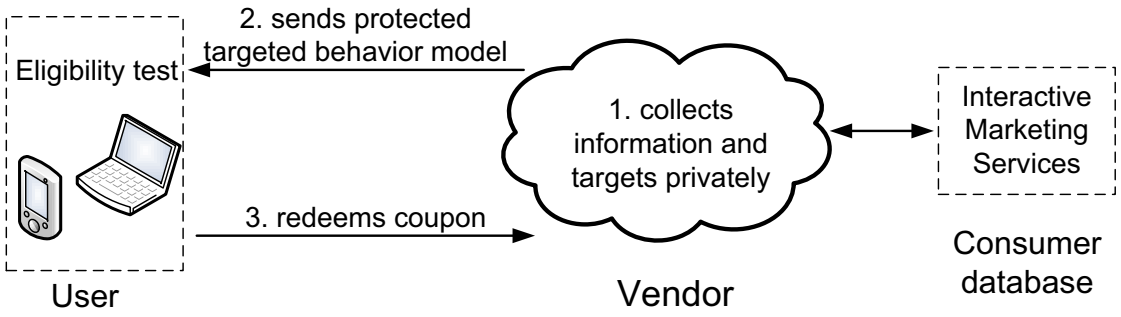


Figure 1: An Exemplary Architecture of Activity-based Targeting

eligible users to prevent coupon exploits. The simpler strategy of pushing down all the coupons in clear and asking the user's device to select ones the user is eligible for does not work well. It exposes the coupons to malicious users who are trying to get discounts they may not have qualified for. Second, a targeting system should ensure that non-eligible users learn nothing about the vendor's targeting strategy (i.e., the coupon's eligibility requirements) beyond their non-eligibility.

Given the complexity of the vendor's targeting strategy, we present two different operating modes for PiCoDa. The first is a non-interactive design that guarantees that behavioral data never leaves a user's device during the coupon targeting process. However, as discussed later in the paper, the non-interactive protocol is only suitable when the vendor's targeting strategy for a coupon is difficult to guess, i.e., its entropy is at least 80 bits. When the vendor's targeting strategy is not hard to guess, we propose a three-round interactive protocol between a PiCoDa server and a user device. In this case, some information about the user's data does leave the client device, but in an indecipherable form that is useless without later cooperation, which the user only provides when they redeem the coupon. We show later in the paper that both designs provide user privacy, vendor protection, and system robustness. Our simulation results with realistic parameter selections further validate the efficiency and effectiveness of our designs.

The rest of the paper is organized as follows. Section 2 introduces the system model, threat model, and our design goals. Then we provide the detailed description of PiCoDa in Section 3. Section 4 gives the security analysis, followed by Section 5, which reports simulation results. Section 6 overviews the related work, followed by our conclusions in Section 7.

2 Problem Statement

2.1 System Model

We consider PiCoDa, the privacy-preserving smart coupon delivery architecture, involves two different entities, as illustrated in Fig. 1: the *user*, who wants to enjoy personalized coupon delivery service while releasing as little as possible private personal data, and the *vendor*, who wants to provide accurate user targeting via behavior analysis while protecting itself from coupon exploitation attacks. In addition, we also assume a *consumer database* in our architecture, which provides proprietary background information (possibly coarse-grained) of consumers to help vendors conduct better targeting services.

The user has a mobile device, which maintains the personal information locally on the device. For simplicity, we assume the user's behaviorial data can be represented as a vector where each entry can be either an integer or real number, denoting representative statistics of different kinds of user behavior over a certain amount of time. For example, these elements in the vector could represent statistics of URL streams in the browsing history, or the number of times specific websites have been visited. The integers could also represent location traces, such as the number of times of different geo-locations that have been visited. The user local data **changes over time**, and provides the most recent targeting information for vendors.

The vendor, who maintains a PiCoDa server for coupon targeting, usually learns some information (e.g., name, mailing address, etc.) about users when they enroll in a loyalty or a coupon delivery program. They may also have a user disclosed profile or can contact commercial consumer databases to learn more (e.g., gender, ethnicity, marital status etc.) about their users. We assume such coarse-grained information is **static** or changes slowly in practice. By combining the static background information and its proprietary behavioral targeting models, a vendor can effectively create the eligibility requirements expressed as targeting strategy for a coupon and only those users that are eligible for the requirements will receive the coupons.

2.2 Behavior Encoding

A vendor may use many different criteria for deciding whether to deliver a particular user a coupon. For example, a vendor may want to reach loyal customers. This is partially reflected by how frequently the user has visited the vendor's store over the past month. Loyalty may also be measured by the user's past purchasing behavior. Or, the vendor may want to reach new potential customers. This might be inferred from visitors to a competitor's store. Vendors may wish to filter prospective coupon recipients according to the probability of the coupon increasing the recipient's future loyalty. This might be informed by a record of coupon deliveries in the past. Finally, physical constraints or inconvenience factors may also be considered. It does not make sense to send a user in New York a coupon redeemable only at California.

In order to accurately answer above questions, vendors must start from the user's behavior raw data and generate/estimate an eligibility strategy $\mathbf{w}$. In order to be effective, $\mathbf{w}$ must be expressive enough to reflect vendor's strategies. In mobile domain, we consider features from four different types of user behavior: browsing history, geographic traces, purchasing information, and message/contact information. For each coupon, frequency and feature counts might be aggregated over several weeks, while the targeting window could cover a shorter period of a few days.

Fig. 2 gives a more detailed list of features that might be used in a typical targeting scenario. A vendor chooses the features and values of the features they wish to target, and assembles them into a vector. For example, to target users that visited the vendor's website at least twice in the past five days, had visited the vendor's retail store three times in the past 10 days, and had made four purchases in the past month, the vector $\mathbf{w}$ might be encoded as $(2, 3, 4)$ for features (w_1, w_2, w_3) where w_1 is number website visits in the past five days, w_2 is the number of retail store visits in the past 10 days, and w_3 indicates the number of purchases in the past month. In our current system, we restrict criteria to approximate or exact equality matching.

2.3 Security Threats and Design Goals

We consider the protection in the PiCoDa architecture design from both user and vendor sides. Both the user's local behavioral information and vendor's targeting strategy (i.e., the proprietary algorithms utilizing the user's up-to-date behavioral data, purchased or collected static data about users and eligibility requirements of a particular coupon) should be protected. In other words, the system should satisfy the following properties:

- **User data privacy**. Our design aims to guarantee that no user behavioral data is revealed to the vendor during the targeted coupon delivery unless the coupon is redeemed. In case the vendor's targeting strategy is complicated and hard to guess, our design further aims to achieve an ideal case in which the communication between user's device and a PiCoDa server is one way: from server to user device.

- **Vendor protection**. The vendor's coupons and its delivery strategy should be protected from non-eligible users during the coupon delivery process. That is, from the information pushed down to user devices, a user either learns he is eligible for a coupon or learns nothing beyond his non-eligibility for that particular coupon.

TARGETING CRITERIA	EXAMPLE METRIC
Browsing History	
overall interest	Fraction of queries in vendor's campaigned product line over all queries in past month
recent interest	Total number of relevant search queries in the last 5 days
recent interest	Total number of visits to product line-relevant webpages in the last 5 days
loyalty	Fraction of webpage views of vendor's product line relative to general web activity
Geolocation Trace Features	
overall interest	Fraction of visits at vendor's retail store versus all shopping stores.
overall interest	Total time in hours the user has spent at vendor's retail stores.
in-market	Time in days since last visit to vendor's retail store
recent interest	Total number of visits to vendor's retail store in the past 5 days
loyalty	Fraction of visits to vendor's retail store versus competitor retail stores
Purchasing Information	
recent purchase volume	Total number of purchases made in the last 30 days
in-market	Total purchases of similar products falling into vendor's product lines.
in-market	Time since the last purchase within vendor's product line
in-market	Number of purchases of complementary products in past 2 weeks
coupon use	Fraction of purchases with coupons over all purchases.
Messaging/Contact Information	
recent interest	Fraction of messages containing keywords related to vendor's product line category in past 5 days
in-market	The number of the user's recent contacts that have purchased within the vendor's product line

Figure 2: Example features for coupon targeting. This list is by no means exclusive or complete. Additional features may be used for improved targeting.

- **Robustness**. The system design should guarantee robust delivery of coupons to eligible users. Moreover, it should prevent users from faking the behavioral data to collect coupons in any illegitimate manner. However, this is an ambitious design goal in the presence of collusion attacks where eligible users share information with non-eligible users. Against such attacks, we propose a series of alternatives to decrease the marginal gain of a coupon, thus discouraging users from arbitrarily trying different behavioral data to maliciously collect and redistribute coupons.

In short, our goal is to enable vendors to deliver behaviorally targeted coupons without accessing any user sensitive information and protect themselves from disclosing any valuable data and against coupon exploitation attacks in the process. However, it is important to note that we do not attempt to protect user's privacy when he actually redeems a coupon due to two reasons: (1) Users willing to redeem the coupon inevitably reveal their eligibility for the coupon at the point of redemption. (2) Vendors need to utilize users' feedback, in terms of the coupon redemption results, to evaluate and improve their targeting strategies.

3 PiCoDa Protocols

In this section,we first discuss expressing coupon eligibility requirements based on the user's behavioral model and the vendor's strategy. Then, we present the details of two PiCoDa protocols for non-interactive and interactive operation.

3.1 Coupon Eligibility Requirement

Following existing literature on behavior targeting (e.g., [11, 3]), we use vectors to represent both vendor side targeting strategy and user side behavioral model. As defined above, the vendor's targeting strategy is represented by an n-dimensional vector $\mathbf{w} = (w_1, w_2, \ldots, w_n)$. Each user's behavioral model, which contains a series of features from the daily behavior events collected by the mobile device, is also denoted by an n-dimensional vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. The targeting process thus depends on the eligibility test between the vendor's strategy $\mathbf{w}$ and the user's model $\mathbf{x}$, and we assume $\mathbf{w}$ and $\mathbf{x}$ are from the same n-dimensional space. Based on different encodings of the user's behavior and the vendor's strategy, the eligibility of a coupon is determined by the following three cases:

1. Entries in $\mathbf{w}$ approximately match entries in $\mathbf{x}$ — in this case, vendors rely on a series of predictive features to distribute coupons. Those features are numerical values computed from user's daily behavior stream. How close $\mathbf{w}$ and $\mathbf{x}$ are is measured by certain distance metrics, such as Euclidean distance and/or cosine distance [8], depending on the different application scenarios.

2. Entries in $\mathbf{w}$ exactly match entries in $\mathbf{x}$ — in this case, vendors rely on a series of deterministic rules to distribute coupons. For example, the vendor may ignore other entries in $\mathbf{w}$ but only care about whether the user has been to a certain local retail store or has been the vendor's loyalty program member to offer him a coupon. Since these rules are usually encoded as discrete binary or categorical values, distance based similarity measurement might no longer be meaningful.

3. Hybrid of case (1) and (2).

For distance based eligibility testing, various data preprocessing techniques, such as min-max normalization [8], can be applied to the derived model vector $\mathbf{w}$ and $\mathbf{x}$. For presentation simplicity, we assume these preprocessing steps are appropriately coordinated between a PiCoDa server and a user device before any coupon delivery takes place.

As discussed in the following sections, our eligibility test only guarantees that the users getting a coupon are eligible for it. However, by design, it is possible for a subset of eligible users not to get a coupon. We argue that small number of false negatives are acceptable to vendors. As long as the number of false positives, i.e., non-eligible users getting a coupon, is negligible, we consider vendor's interest not to be violated. The rest of this section shows how we leverage these assumptions in the design of PiCoDa.

3.2 Protocol 1: Privacy-preserving Non-interactive Coupon Targeting

In non-interactive protocol, the PiCoDa server pushes down the targeting strategy in a protected form to the user, who then performs a blind matching with local behavioral model to determine his eligibility status for a coupon. Such a design has the benefit that nothing leaves the user's device before the user actually gets a coupon. An assumption in the non-interactive targeting design is that the vendor's targeting strategy must be hard to guess, i.e., its entropy should be at least 80bits, to be secure against brute-force guessing attacks. Otherwise, a malicious user can find a behavioral model that matches vendor's strategy by trial and error.

Compared to having rules that dictates exact match on few variables, high entropy requirement for targeting strategy can be more easily met if $\mathbf{w}$ contains many predictive features with numerical values. Thus, we focus on the first case of eligibility test, where a user's eligibility for a coupon is determined by measuring the similarity between the vendor's targeting strategy $\mathbf{w}$ and the user's behavioral model $\mathbf{x}$. As noted previously, the eligibility test has to happen at the user's device for user data privacy and better scalability.However, the targeting strategy $\mathbf{w}$ cannot be pushed down to the user device in clear as it would violate two of previously mentioned design goals: vendor protection and robustness. To achieve the challenging goal of privacy-preserving non-interactive coupon targeting, we use locality sensitive hash functions to implement private similarity tests.

Locality Sensitive Hashing. Locality-sensitive hashing (LSH) denotes the method to perform probabilistic dimension reduction of high-dimensional data [1]. Its key idea is to hash the input data points (using specially-designed locality-sensitive hash functions), such that for similar data points (close to each other), the collision probability is much higher than for those that are far away. For different distance metrics, the family of LSH is defined differently [1]. For presentation simplicity, we take the LSH defined over cosine distance as an example in our targeting scheme description, though LSH for other distance metrics, such as Euclidean distance [5] or Hamming distance [1] etc., can also be used.

The cosine distance metric can be represented by the angle between two vectors $\mathbf{w}$ and $\mathbf{x}$, $\Theta(\mathbf{w}, \mathbf{x}) = \cos^{-1}(\frac{\mathbf{w}\cdot\mathbf{x}}{||\mathbf{w}||\cdot||\mathbf{x}||})$.

For this distance measure, Charikar [4] gives the following LSH family $\mathcal{F}$. By drawing each component of an n-dimensional random vector $\mathbf{r}$ from the Gaussian distribution $\mathcal{N}(0, 1)$ independently, the hash function $f_{\mathbf{r}}(\cdot)$ computed over an n-dimensional vector $\mathbf{q}$ is given by:

$$f_{\mathbf{r}}(\mathbf{q}) = \begin{cases} 1 & \text{if } \mathbf{r}\cdot\mathbf{q} \geq 0 \\ 0 & \text{if } \mathbf{r}\cdot\mathbf{q} \leq 0 \end{cases}$$

This construction divides the entire input space of the dataset by the hyperplane represented by the vector $\mathbf{r}$; two vectors lying on the same side of the hyperplane defined by $\mathbf{r}$ hash to the same value. The likelihood of two vectors $\mathbf{w}$ and $\mathbf{x}$ hashing to the same LSH value depends on their cosine similarity, i.e.,

$$p = \Pr[f_{\mathbf{r}}(\mathbf{w}) = f_{\mathbf{r}}(\mathbf{x})] = 1 - \frac{\Theta(\mathbf{w}, \mathbf{x})}{\pi}. \tag{1}$$

As using one hash function $f_{\mathbf{r}}$ from the family $\mathcal{F}$ does not give accurate enough results for the locality sensitive hash, in practice, it is suggested to use a set of K hash functions $f_{\mathbf{r}_1}, \ldots, f_{\mathbf{r}_K}$, and the final hash value is obtained by concatenating their output. This K-bit LSH function, denoted as $F(\cdot)$, maps an n-dimensional vector $\mathbf{x}$ into a K-bit string.

Non-interactive Protocol for Approximate Matching. However, due to the locality sensitivity, where similar vectors will be hashed together, LSH no longer has the one-way property of a cryptographic function. In other words, it is possible for adversaries to infer information on the pre-image of LSH from the locality sensitive hash values. To enhance the security strength, we propose to apply a cryptographic hash function, like SHA1, to the locality sensitive hash values, before the targeting.

Specifically, instead of pushing down the value of $F(\mathbf{w})$ and asking the user to compare $F(\mathbf{w}) \stackrel{?}{=} F(\mathbf{x})$, the PiCoDa server can push down $h(F(\mathbf{w}))$ and $F(\cdot)$ to the user. The user computes $F(\mathbf{x})$ and tests if $h(F(\mathbf{w})) \stackrel{?}{=} h(F(\mathbf{w}))$. If the test matches, the user is potentially eligible for a coupon to redeem. If not, then due to the one way property of $h(\cdot)$, the user will learn nothing about the $F(\mathbf{w})$ from the received hash values. The non-interactive protocol between the PiCoDa server and the user is as follows. We let $\phi(\cdot)$ denote a pseudorandom function, $\mathsf{Enc}(\cdot)$ denote a semantically secure encryption function, and $\mathsf{Sig}(\cdot)$ denote some secure digital signature scheme.

1. The PiCoDa server sends down to the user:
   $h(F(\mathbf{w}))$, $\mathsf{Enc}_{\mathsf{key}}(coupon||UID||nonce||\ Sign)$, and $F(\cdot)$. Here $coupon$ denotes the actual content of coupon, UID specifies the user, $nonce$ is a fresh random number per hash value pushed down. Also $\mathsf{key} = \phi(F(\mathbf{w}))$, $Sign = \mathsf{Sig}_{vendor}(coupon||UID||nonce)$.

2. The user tests his behavioral data and checks if $h(F(\mathbf{x})) \stackrel{?}{=} h(F(\mathbf{w}))$. If yes, the user continues to plug $F(\mathbf{x})$ into $\phi(\cdot)$ to get the trapdoor key, and further open the encrypted coupon. If no match, the user learns nothing beyond the fact of his non-eligibility, due to the assumption that $\mathbf{w}$ is hard to guess.

3. During redemption, the validity of a coupon can be checked by verifying the signature, $Sign$, and the UID of the redeeming user.

Parameter Selection. Because $F(\mathbf{w})$ outputs K-bit string, we have to ensure K is sufficiently large, e.g. K = 80, such that it is not feasible for malicious users to enumerate. However, larger K value would also reduce the probability of two similar points hashing together (see Eq. (1)), due to the fact that $p > p^K$ for any $0 < p < 1$ and $K > 1$. Since the success of the eligibility test depends on the similarity of the two vectors $\mathbf{w}$ and $\mathbf{x}$, setting large K might result in less or even no successful matches.

Following the methodology in LSH community [1], to maintain the correctness of the high probability matching, one approach is to push down a set of L independent concatenated LSH functions $F_1(\cdot), \ldots, F_L(\cdot)$, and ask the user to find if any of the L hash values matches his own result. Note that the probability for the user to find any match among the L hash values is at least $1 - (1 - p^K)^L$, where p is determined by the similarity of $\mathbf{w}$ and $\mathbf{x}$ via Eq. (1). Clearly, by increasing L, we increase the value of $1-(1-p^K)^L$, and thus maintain the high probability matching for true positive of the eligibility test. By increasing K, we decrease the value of $1-(1-p^K)^L$, and thus suppress the low probability matching for the false positive. As a result, choosing large K and L amplifies the gap between the true positive and false positive of the eligibility test. However, the side-effect is the extra computation burden at the user side.

In practice, because only the seed used to randomly sample the vectors for function $F(\cdot)$ needs to be sent, the bandwidth for transmitting multiple LSH functions is not a concern. To avoid intensive computation cost for the eligibility test, the PiCoDa server can distribute the L hash values over a certain targeting time window instead of in one batch. For example, it can push down 10 different hash values on the same targeting vector $\mathbf{w}$ with 10 randomly different LSH functions every day to one user in a 2-week time window to reach the requirement of $L = 140$. If the user behavioral model matches any of those hash values, an eligible coupon is to be delivered. Otherwise, after this targeting time-window, the PiCoDa server can start a new cycle and push down hash values based on some different targeting strategy.

Remark. When the PiCoDa server pushes down $h(F_i(\mathbf{w}))$ to each user each time, where $i = 1, \ldots, L$, it must ensure that every $F_i(\mathbf{w})$ are at least have 1 bit difference. As a result, it can ensure $\mathsf{key} = \phi(F_i(\mathbf{x}))$ is only usable for that specific user with UID and the specific coupon with $nonce_i$. Since $F_i(\cdot)$ are defined by random vectors, even for the same $\mathbf{w}$, making each $F_i(\mathbf{w})$ different should not be difficult to achieve in practice. For ease of presentation, we defer the security analysis to Section 4.

3.3 Protocol 2: Privacy-preserving Interactive Coupon Targeting

As discussed in Section 3.1, there are cases in which the vendor's strategy is deterministic instead of approximate. For example, the vendor may only care about whether the user has been to a certain local retail store to distribute him a coupon. Further, such deterministic rules are usually not complicated, i.e., do not have high enough entropy. This may be the result of that vendors don't care about certain entries in strategy $\mathbf{w}$ or because in certain scenarios only a few entries matter for the targeting purposes. Thus, directly pushing down the hash values $h(\mathbf{w})$ (due to deterministic match, we don't need LSH any more) to users for local matching no longer works, as simple guessing attacks via enumeration on value of $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ become feasible.

To cover this case, we propose a design by making certain relaxations of our stringent constraints. That is, our protocol now requires users and the PiCoDa server to interact during the coupon delivery session. But we still ensure that the vendor's strategy is protected against non-eligible users, and users behavioral data is not revealed to the vendor unless they choose to redeem the coupon (if they are eligible).

Assuming the vendor only cares about m entries in $\mathbf{w}$ with index $\mathcal{I} = (i_1, i_2, \ldots, i_m)$. In the following, we adopt techniques from "Password-authenticated key agreement" [2] as a base for our protocol design. Let $\mathbb{G}$ denotes a finite cyclic group with generator g. This group could be $\mathbb{Z}_P^*$ where P is a large prime

with 1024 bits. Both g, P and hash function $h(\cdot)$ are public. The interactive protocol of PiCoDa operates as follows:

1. The PiCoDa server picks random values $r, a \in \mathbb{Z}_P^*$, computes $H_v = h(w_{i_1}||w_{i_2} \dots w_{i_m}||r)$ and $g^a \mod P$, and sends $\{\mathsf{Enc}_{H_v}(g^a), r, \mathcal{I}\}$ to the user.

2. The user picks $\{x_i\}$ vis $\mathcal{I}$ and computes $H_x = h(x_{i_1}||x_{i_2} \dots x_{i_m}||r)$. He picks a random $b \in \mathbb{Z}_P^*$, computes $g^b \mod P$, and sends $\mathsf{Enc}_{H_x}(g^b)$ to the PiCoDa server.

3. The PiCoDa server uses H_v to decrypt $\mathsf{Enc}_{H_x}(g^b)$ and gets decrypted value V. It then sends $\mathsf{Enc}_{\mathsf{V}^a}(coupon||UID||nonce_i||Sign_i)$ to the user.

4. The user uses H_x to decrypt $\mathsf{Enc}_{H_v}(g^a)$ and gets decrypted value X. He then uses X^b to decrypt $\mathsf{Enc}_{\mathsf{V}^a}(coupon||UID||nonce_i||Sign_i)$.

Note that in step (3) and (4), after decryption, we have $\mathsf{V} = g^b$ and $\mathsf{X} = g^a$ if and only if $H_v = H_x$. Otherwise, both V and X are just some indistinguishable random values. In step (4), when $H_x = H_v$, it's easy to know $\mathsf{X}^b = \mathsf{V}^a = g^{ab}$. And the eligible user gets the coupon in the final step. In the meanwhile, this eligible user knows his x_i for $i \in \mathcal{I}$ equals vendor's corresponding w_i. However, when $H_x \neq H_v$, non-eligible users still know nothing about vendor's strategy. The detailed security analysis in Section 4.

Remark. To prevent users faking their behavioral data –by colliding with eligible users who received the coupon before them–, PiCoDa server may collect commitments from user devices to their behavior vectors (e.g., $h(x_i||i||UID||nonce)$) and require them to open the commitments while redeeming a coupon (e.g., by revealing the $nonce$). Alternatively, PiCoDa server may run the first 2 steps of the above protocol with all users before executing step 3 with any of them, or hide the coupon encryption key in step 3 (e.g., by using $h(\mathsf{V}^a||nonce)$ as the encryption key) and open it (e.g., by revealing the $nonce$) to all the users at the same time. Finally, we remark that in practice, the PiCoDa server initiates this protocol with each user only once per coupon, which practically excludes the threat of coupon exploit from malicious users by limiting guessing and exhaustive search opportunities.

3.4 Further Discussion: Dealing with the Hybrid Case

We have discussed the vender's approximate strategy and deterministic strategy. What if the vendor needs both? One way to achieve that is to concatenate the two protocols. In particular, the vendor first uses the non-interactive protocol to do the LSH based targeting. The eligible users passing the test have the choice of whether to proceed to do the interactive protocol or not. This concatenation can be applied to the case of tiered coupon distribution systems, where approximate matching corresponds to loose eligibility requirements and the vendor delivers broadly targeted coupons, like $1-off for a sports retailer. When the user chooses to proceed for interactive protocol, it comes to a more personalized targeting. For example, such coupons can be for only a few users that are very important and highly loyal to the vendor. Of course, with the coupon becomes more personalized or higher-tiered, the users are willing to reveal more of themselves (when redeeming the coupon).

Instead of doing tiered coupon delivery, the vendor might be only interested in offering coupons if and only if both the approximate and the deterministic strategies have positive matches simultaneously. In this case, we can put both LSH values and the deterministic rules in the cryptographic hash $h(\cdot)$ in the step (1) of either the non-interactive protocol[1] or the interactive protocol, the remaining of the protocols follows directly. Take the interactive design for example. Let $H_v = h(w_{i_1}||w_{i_2} \dots w_{i_m}||F(\bar{\mathbf{w}})||r)$, where $\bar{\mathbf{w}}$ contains the $n - m$ remaining entries of the original $\mathbf{w}$ seeking for approximate match. Because the

[1]The combined strategy of the non-interactive protocol must have an entropy larger than 80bits.

interactive protocol does not allow users/PiCoDa server to do offline guessing/enumeration attack, we no longer need the 80-bit requirement for the LSH outputs. In other words, we only choose an appropriately small number of LSH output bits K such that we can use $L = 1$ to simplify the eligibility test. The protocol goes exactly the same as in Section 3.3 and thus is omitted.

Remark. From the ease of management point of view, the non-interactive mode of PiCoDa is easier to operate since all the targeting hash values could be pre-generated. The PiCoDa server does not even have to be always online as there are no interactions. For the interactive operation mode, the PiCoDa server needs to interact with every user per coupon delivery, which can be less scalable than the non-interactive case. However, it does give the vendor more flexibility when choosing targeting strategies. In both cases, users maintain the full control of their behavioral data until they redeem the coupons. Non-eligible users know nothing about the vendor's targeting strategy.

4 Security Analysis

4.1 User Data Privacy Protection

Non-interactive Coupon Targeting. User privacy is protected in the sense that all the eligibility matching happens at user's mobile device and no data leaves the phone before the user redeems the coupon. However, when a user decides to redeem the coupon, he or she must disclose to the PiCoDa server his or her eligibility status. In this case, the vendor can learn that $F(\mathbf{w}) = F(\mathbf{x})$, where $F(\cdot)$ is the public locality sensitive hash function.

Though we limit the privacy-preservation to the targeting process, it is worth further understanding on how much the fact $F(\mathbf{w}) = F(\mathbf{x})$ reveals about $\mathbf{x}$. Since we choose $K = 80$, which divides the whole n-dimensional space into 2^{80} subspaces. Thus, if there are enough reasonable points in the same subspace, then user's $\mathbf{x}$ can further be protected in a k-anonymity manner. If in the worst case there is only one point in the subspace, then the vendor can exactly pinpoint $\mathbf{x}$ via $\mathbf{x} = \mathbf{w}$. However, since the user knows the subspace as well, the user can also exactly pinpoint vendor's $\mathbf{w}$ from the eligibility test, which violates vendor's own protection requirement. So we argue that the vendor has enough incentives not to select small subspace so as to protect the targeting strategy $\mathbf{w}$. As a result, whenever a user finds a match, his behavioral data $\mathbf{x}$ sharing the same subspace with $\mathbf{w}$ will also be protected from that same large subspace.

Interactive Coupon Targeting. In this case, users exchange information with the PiCoDa server. But based on the security strength of "password-authenticated key agreement" [2], we still ensure that users have full control of their behavioral data. First, information uploaded to the PiCoDa server in the protocol is just some random encryption value; Second, even after the decryption, the PiCoDa server cannot tell whether the user's $\mathbf{x}$ matches the strategy $\mathbf{w}$. In other words, before the coupon redemption, the PiCoDa server or vendor learns nothing about the coupon targeting result. Thus, user data privacy is well-protected.

4.2 Vendor Protection

Non-interactive Coupon Targeting. By protecting vendor, we aim to ensure the eligibility test on user's device either reveals the fact to user that $F(\mathbf{w}) = F(\mathbf{x})$ or nothing about vendor's targeting strategy $\mathbf{w}$ except that $F(\mathbf{w}) \neq F(\mathbf{x})$. For the latter, due to our two-layered hash construction with large $K = 80$, the user knows nothing from his unmatched eligibility test. This is because reverse-engineering $h(F(\mathbf{w}))$ is computationally infeasible, assuming $\mathbf{w}$ itself is hard to guess, i.e., with 80-bit entropy.

But if there is a match, then the user knows $F(\mathbf{w}) = F(\mathbf{x})$. Using the aforementioned argument where the vendor selects a large enough subspace defined by random vectors for $F(\cdot)$, the vendor's $\mathbf{w}$ cannot be exactly pinpointed by a single user.

Interactive Coupon Targeting. In the interactive case, there are only a small number of rules or entries in the targeting strategy $\mathbf{w}$, or the vendor selectively cares a portion of entries in $\mathbf{w}$. Thus, protecting both the interesting entry index and the values can become important to vendor. Currently, our design does not

protect which entries are important in $\mathbf{w}$. Knowing this information might give the user some advantage to infer the actual values in $\mathbf{w}$, based on other context information. However, the vendor can instruct the PiCoDa server to initiate the protocol with each user per coupon only once, and thus each user only has one chance to guess the correct value in vendor's targeting strategy $\mathbf{w}$. From a practical point of view, the threat of correct guessing and other coupon exploits can be negligible. Further, following the same reasoning for user privacy protection, we can ensure that non-eligible users know nothing about the actual values in $\mathbf{w}$ from the eligibility test.

Remark. Note that neither design maintains the vendor protection against users having a match from the eligibility test. A positive matching result inevitably reveals some information about the vendor's strategy to the users. Ideally the vendor would prefer not to expose the strategy at all. To mitigate the negative effect of exposing targeting strategies to eligible users, we propose a series of alternative approaches in the next section.

4.3 Robustness

Previous discussions show that users who are originally not eligible for a coupon learn nothing beyond the failure of the eligibility test, and thus are not able to provide useful information to harm the system. However, eligible users who already get the coupons might be willing to share information of their behaviors, e.g., via blogs, or social networks to their friends. These users might give good pointers for other users to mimic the behavior and narrow down the brute-force guessing space directly on $\mathbf{x}$ for $\mathbf{w}$ or $F(\mathbf{w})$. Unfortunately, there is no perfect solution for the vendor to defeat a user that is faking his behavior. In the following, we provide a series of alternatives to address the problem. Our goal is to prevent or discourage users from arbitrarily trying different behavior $\mathbf{x}$ to maliciously collect and redistribute coupons.

Using Trusted Computing Technology. Our first approach is to rely on trusted computing technology to mitigate the concern of user's faking behavior. It can be achieved via Trusted Platform Module (TPM) [14], which offers hardware based root of trust and has already been adopted by many major laptop vendors in the market. Physically attached to a computer, the TPM chip is accessed by software from upper layers using a well-defined command set, through which, the TPM can facilitate cryptographic functionalities like hardware pseudo-random number generation, key generation, signing and encryption/decryption etc. Thus, we can use TPM's capabilities to do code attestation and verification for the device and the application software.

According to the latest work-in-progress specification version 2.0 of Mobile Trusted Module by the TCG [13], TPM is expected to be soon in place on smart phones from major phone manufacturers. Assuming users cannot temper the process running on device collecting user's behavioral data, then users have to actually conduct the behavior accordingly to get the coupon, like visiting the stores, or accumulating enough purchase records. The marginal gain of coupon can thus be easily diminished by the cost of non-eligible user's actually mimicing/conducting those possibly non-trivia behaviors.

Commitment Based Approach. We can also ask users to commit to their behavioral data $\mathbf{x}$'s periodically or before receiving coupons. In this case, they cannot arbitrarily change their behavior to maliciously collect coupons, even if they learn information by colluding with each others. In Section 3.3, we have outlined few commitment based approaches for interactive targeting of PiCoDa. In the following, we demonstrate another example via using Pedersen's commitment [10] scheme for non-interactive targeting mode of PiCoDa.

Assume both the vendor and the user agree on some group $\mathbb{G}_q$ of prime order q and two generators g, g_0 for which the discrete logarithm problem is hard. Whenever the user conducts certain behaviors, represented by some element x_i in the behavioral model $\mathbf{x}$, the user picks a random $\tau_i \in \mathbb{Z}_q$ and sends a commitment $C_{\tau_i}(x_i) = g_0^{\tau_i} g^{x_i}$ to the PiCoDa server. Here due to the randomness of τ_i, x_i is protected. Given commitments $C_{\tau_i}(x_i)$ for $i = 1, \ldots, n$, the PiCoDa server could later verify the result of vector product $\mathbf{r} \cdot \mathbf{x}$ from the LSH computation (See Section 3.2), based on the homomorphic property of the commitment

construction. Specifically, the user redeeming the coupon sends $\mathbf{r} \cdot \mathbf{x}$ together with the randomness $\{\tau_i\}$ embedded in the commitment. The PiCoDa server verifies $\prod_{i=1}^{n} C_{\tau_i}(x_i)^{r_i} = g^{\mathbf{r}\cdot\mathbf{x}} g_0^{\sum r_i \cdot \tau_i}$ and thus check if the corresponding bit of LSH output is correct.

Compared to TPM based approach, commitment based approach requires the user to send commitments to the PiCoDa server, which could be against the original motivation of a non-interactive targeting design. However, commitments do not have to be done very frequently, because $\mathbf{x}$ values are usually aggregate information over a certain amount of time. If the coupon targeting time-window is set to be 2 weeks, i.e., $\mathbf{x}$ measures the user's behavior over the past 2 weeks, then asking users to send non-revealing information to the PiCoDa server once per 2 weeks can be reasonably acceptable .

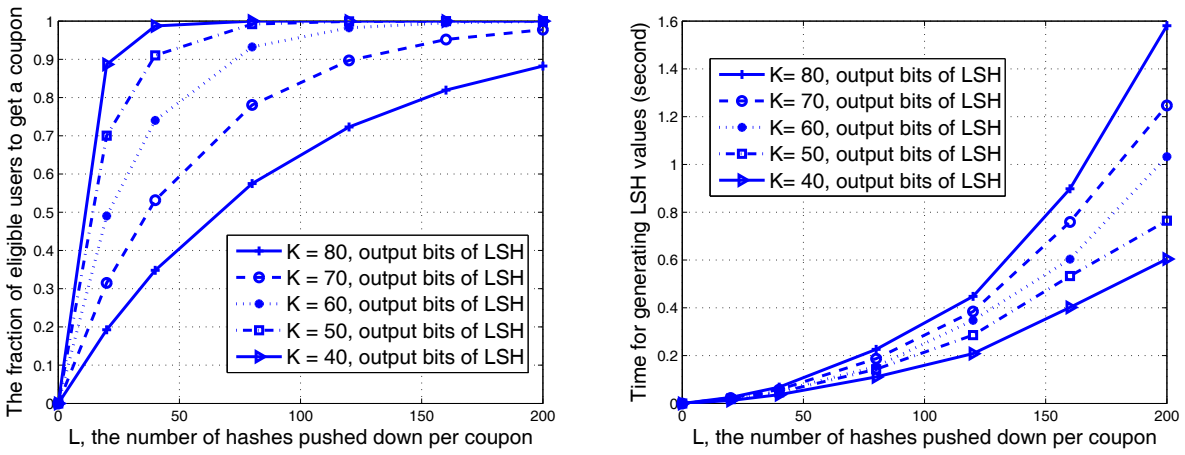
Relying on External Third Parties. In practice, we can also rely on external third parties to help prevent users from faking behaviors. The immediately available third party for the role could be the wireless carriers. The carriers keep a track of their mobile users's geographic locations all the time. One viable approach is to have vendors and wireless carriers setup some service agreement such that periodically vendors can rely on wireless carriers to verify users' location data. Thus, all the geographic related behavior can be verified by vendor. Specifically, when a user has received coupons from the proposed PiCoDa protocols and decides to redeem one, he can give the permission (e.g., request with his signature) to the vendor for verifying their geographic traces at the carrier's. Note that because the user's eligibility for the coupon is inevitably revealed to the vendor at the time of redemption (see Section 2.3), the fact that the vendor verifies the user's behavior authenticity through external third parties is not a violation of PiCoDa's design goals on user data privacy protection. Considering a large potion of elements in behavioral model $\mathbf{x}$ might be location related, users only need to prove or commit on other non-location related behaviors and save the computation and bandwidth cost. Following the same intuition, other similar third parties might include: central ad-network dealers, like Google, Yahoo! for helping verifying user's browsing behaviors. Mobile apps platform holders like Apple and Google could help verify user's app-related behaviors.

Relying on Probabilistic Matching Property. Another factor we should take into consideration for discouraging users from faking their behaviors is the probabilistic matching. By selecting appropriate parameters of K and L, the vendor can fine-tune the probability of the successful matching between $\mathbf{w}$ and $\mathbf{x}$ via LSH. For example, even if $\mathbf{w}$ and $\mathbf{x}$ are quite close with each other such that the angle $\theta(\mathbf{w}, \mathbf{x})$ normalized by π is just 0.1, choosing $K = 80$ and $L = 200$ could still leads to a successful match with probability as low as 0.043. This means for 1000 users who are potentially eligible for a coupon represented by a targeting strategy $\mathbf{w}$, the eligibility test only gives coupons to at most 43 users on average.

As mentioned in Section 3.1, having a relatively small fraction of eligible users get the coupons is not violating vendor's business interest. Therefore, for those users who originally do not have the correct behavioral data and want to learn information shared by others to try their luck, the slim rate of successful matching can be really discouraging for their motivation of collusion.

Relying on Coupon Redemption History. Each time a user redeems a coupon, the vendor knows the user's then behavioral model $\mathbf{x}$ is within a certain distance of the eligibility model $\mathbf{w}$ where $\theta(\mathbf{x}, \mathbf{w}) < d$. Over time, such relationship $\theta(\mathbf{x}, \mathbf{w}) < d$ might reveal a certain pattern. Exploring the common patterns from those relationships can help the vendor identify the inconsistencies of user's behavioral models in consecutive coupon matching/redeeming sessions over a enough long time period. Another approach is to combine the coupon redemption pattern with aforementioned commitment schemes. For example, if a user has redeemed a 5% coupon over the past three coupon delivery cycles, and then suddenly wants to redeem a coupon for "buy 1 with 1 free" that can be suspicious. The vendor could honor the coupon this time but start to request the user's behavior commitments for future eligibility verification. The more coupons a user has redeemed, the more difficult for the user to fake things.

Remark. Due to space limitation, we do not try to enumerate a comprehensive list and believe there could



(a) The eligibility fraction of targeting.

(b) The user side computation cost.

Figure 3: User eligibility and LSH generation times in the non-interactive design, for different choices of K and L.

be other options available as more research effort is put on the topic. We argue that putting together all the listed alternatives or operating them in parallel, where some of them can be overlapping, could significantly raise the bar for unfaithful users. Also, the overall effect for the proposed PiCoDa system for private coupon targeting is much better than the current simple coupon code based ecosystems, in terms of ensuring user data privacy, vendor protection as well as system robustness.

5 Performance Evaluation

We evaluate PiCoDa through simulation to validate the running times for realistic parameter values. Both mechanisms of PiCoDa are implemented in C++ on a workstation with Intel Core 2 CPU running at 3.0GHz. The OpenSSL library is used to implement cryptographic functions like SHA1 and AES etc.

In a typical targeting scenario, the vendor's targeting strategy may cover the user's behavior events from domains like page views, query search results, GPS traces, purchase history, messages, and contacts. And for the behavior events from each domain, there can be a series of measurements to be reported as features in $\mathbf{w}$ and $\mathbf{x}$. In our simulation, without loss of generality we set the dimensionality of user's behavioral model and vendor's targeting strategy $|\mathbf{w}| = |\mathbf{x}| = 30$. The cosine distance-based similarity measurement is used, and the hash based commitment scheme is included in the design to ensure that no user could fake their behavioral data. Note that we only report timing performance for the protocol data. A practical implementation will require time to transmit the coupon contents.

Non-interactive Design: In this case, we fix the cosine distance threshold between $\mathbf{x}$ and $\mathbf{w}$ at 0.985, which means the largest tolerated angle between $\mathbf{w}$ and $\mathbf{x}$ is $\theta(\mathbf{x}, \mathbf{w}) = 18.2°$. The results for different choices of K and L are shown in Fig. 3. Fig. 3-(a) shows the fraction of eligible users that receive a coupon after the eligibility test. Depending on the application, this can be fine-tuned by the vendor by setting K and L appropriately. In particular, given any fixed K ranging from 40 to 80,[2] the vendor can always find some L less than 200 such that the fraction of eligible users that do receive a coupon is more than 90%.

On the other hand, K and L cannot be set arbitrarily, as high values might increase the computational burden on the user side for each coupon targeting as shown in Fig. 3. In our simulation, to avoid the transmission of LSH function, which is defined by a set of random vectors and can be large for large K and

[2]For K less than 80, we implicitly assume a hybrid case such that deterministic rules are combined with LSH output to satisfy the 80-bit entropy requirement of the targeting strategy.

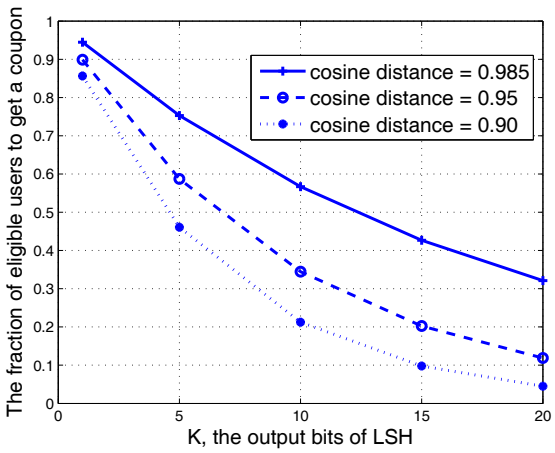


Figure 4: The eligibility fraction of interactive targeting for different choices of K.

L, only a random seed must be transmitted from the PiCoDa server to user, who then generates the LSH function on the fly. The timing result reported in Fig. 3-(b) thus involves both LSH function generation and LSH value computation. It can be seen that timing cost increases when either K or L is large. But even for the largest K and L values on the graph, the computation still requires less than 1.6 seconds. As coupon delivery does not need to happen in real time, this computational cost is likely to be acceptable in practice.

Interactive Design We also simulate the hybrid case in Section 3.4 using the interactive protocol of PiCoDa. As we no longer have the constraint on the large number output bits K of LSH, we can set $L = 1$ and choose an appropriate K value to fine-tune the acceptable accuracy with regard to different datasets. The fraction of eligible users that get a coupon for three different thresholds is shown in Fig. 4. Under those settings, i.e., $K < 20$, the computation cost for generating and evaluating LSH values is always less than 1 millisecond. Also, the three-round interaction between the PiCoDa server and the user is very efficient. Each step only involves one modular exponentiation together with AES encryption and hash operations, which takes less than 1.5 milliseconds.

Note that although our timing results are derived from simulation on a desktop machine, it is reasonable to expect that mobile devices will match this performance in the next few years, given the current trend of increasing mobile device processing power. We leave the empirical study of PiCoDa with a real dataset on mobile devices as future work.

6 Related Work

Privacy-preserving targeted behavior analysis has been explored by researchers in various forms [12, 7, 6]. Toubiana *et al.* proposed Adnostic [12], a browser extension that runs the behavioral profiling and targeting algorithm on the user browser's history database. Because the results are kept within the browser, users see ads relevant to their interests from a group of candidate ads (they suggest 20) without leaking information outside the browser. Adnostic uses homomorphic encryption and zero-knowledge proofs to allow the ad-network to correctly charge the corresponding advertisers, without seeing which ads are viewed by users (i.e., the so-called "charge per impression" model). Adnostic does not consider it a privacy breach when users reveal their ad click history. This is similar to our PiCoDa system, as we don't aim to protect user privacy when the user chooses to redeem the coupon. What differentiates PiCoDa and Adnostic is the security requirements. Adnostic only considers the user's privacy, while our PiCoDa system further ensures vendor protection, and enforces the eligibility test and coupon result validation for system robustness.

Guha *et al.* present an architecture called Privad [7], which has similar goals of Adnostic but aims to

provide better privacy guarantees of user's local data. Specifically, Privad introduces a semi-trusted *dealer* between the ad-network and user in order to anonymize the user click behavior to prevent the ad-network from identifying the user. The report of a view/click still allows the ad-network to bill the advertisers and pay the publishers accordingly. Though Privad provides better privacy protection than Adnostic, the utilized anonymization mechanism also increases the cost for both performance and the click-fraud detection. As with Adnostic, the difference between Privad and PiCoDa is that Privad does not consider vendor side protection and does not perform the eligibility check during the ad targeting.

Fredrikson *et al.*'s RePriv [6] presents another in-browser approach to perform personalization without sacrificing user privacy. Unlike Adnostic and Privad, RePriv does not hide all the user's personal information. Rather, RePriv shifts the privacy control to the user, i.e., it explicitly asks the user's consent in any transfer of sensitive local information to different service providers for personalized content. While RePriv allows a wide range of personalized web applications to exist, shifting the control of personal information transfer also raises usability concerns over frequent interruptions and the difficulty of specifying preferences about personal information dissemination. PiCoDa instead adopts a different disclosure model. In particular, PiCoDa protects user data during or after the targeting process, unless the user chooses to redeem the coupon. Other differences include the enforced eligibility test and the vendor protection in our system.

7 Concluding Remarks

In this paper, we have studied the problem of privacy-preserving coupon targeting. Our goal is to enable vendors to deliver targeted coupons to eligible mobile users without compromising user privacy and without revealing their targeting strategies. The design of PiCoDa shifts the targeting from vendor side to user side. Specifically, for different vendor targeting strategies, we have provided two targeting protocols: a non-interactive one and a three-round interactive one. Our security analysis shows how both meet the system requirements of user privacy, vendor protection, and robustness. The timing performance from our simulation with realistic parameter selections further validates the efficiency and effectiveness of PiCoDa. Given the results, we conclude that PiCoDa extends existing work on privacy-preserving targeted advertising, which only considers user privacy but ignores vendor protection. Furthermore, we hope PiCoDa will inspire other privacy-preserving targeting services in which both the vendor protection and user privacy protection are demanded.

References

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51:117–122, 2008.

[2] M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attacks. In *Proc. of EUROCRYPT*, pages 139–155, 2000.

[3] M. Bilenko and M. Richardson. Predictive client-side profiles for personalized advertising. In *Proc. of ACM SIGKDD*, 2011.

[4] M. Charikar. Similarity estimation techniques from rounding algorithms. In *Proc. of the 34th Annual ACM Symposium on Theory of Computing*, 2002.

[5] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. of STOC*, pages 253–262, 2004.

[6] M. Fredrikson and B. Livshits. Repriv: Re-envisioning in-browser privacy. In *Proc. of IEEE Symposium on Security and Privacy*, 2011.

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

[8] J. Han, M. Kamber, and J. Pei. *Data mining: concepts and techniques*. Morgan Kaufmann Pub, third edition, 2011.

[9] K. Partridge and J. Begole. Activity-based advertising. In J. Müller, F. Alt, and D. Michelis, editors, *Pervasive Advertising*. Springer-Verlag, London, UK, 2011. to appear.

[10] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proc. of CRYPTO, volume 576 of LNCS*, pages 129–140, 1991.

[11] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proc. of WWW*, 2004.

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

[13] Trusted Computing Group. MTM 2.0 - Trusted Computing Group. Online at `http://www.trustedcomputinggroup.org/`.

[14] Trusted Computing Group. TPM Main Specification. Online at `http://www.trustedcomputinggroup.org/`.

[15] J. Turow, J. King, C. Hoofnatle, A. Bleakley, and M. Hennessy. Americans reject tailored advertising and three activities that enable it. *Departmental Papers (ASC)*, page 137, 2009.

[16] Y. Xu, B. Zhang, Z. Chen, and K. Wang. Privacy-enhancing personalized web search. In *Proc. of the 16th International World Wide Web Conference*, 2007.