Privacy Challenges in Online Targeted Advertising

Minh-Dung Tran

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministérial : 7 août 2006

Présentée par

Minh-Dung Tran

Thèse dirigée par **Dr. Claude Castelluccia**
et codirigée par **Dr. Mohamed-Ali Kaafar**

préparée au sein de **l'INRIA Rhônes-Alpes, équipe Privatics**
et de **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Privacy Challenges in Online Targeted Advertising.

Thèse soutenue publiquement le **13 Novembre 2014**,
devant le jury composé de :

Pr. Martin Heusse
Grenoble INP - Ensimag, Président
Dr. Paul Francis
Max Planck Institute for Software Systems, Rapporteur
Dr. Marc-Olivier Killijian
CNRS, Rapporteur
Dr. Vincent Toubiana
CNIL, Examinateur
Dr. Claude Castelluccia
Inria, Directeur de thèse
Dr. Mohamed-Ali Kaafar
Inria, Co-Directeur de thèse

Abstract

In modern online advertising, advertisers tend to track Internet users' activities and use these tracking data to personalize ads. Even though this practice - known as *targeted advertising* - brings economic benefits to advertising companies, it raises serious concerns about potential abuses of users' sensitive data. While such privacy violations, if performed by trackers, are subject to be regulated by laws and audited by privacy watchdogs, the consequences of data leakage from these trackers to other entities are much more difficult to detect and control. Protecting user privacy is not easy since preventing tracking undermines the benefits of targeted advertising and consequently impedes the growth of free content and services on the Internet, which are mainly fostered by advertising revenue. While short-term measures, such as detecting and fixing privacy leakages in current systems, are necessary, there needs to be a long-term approach, such as privacy-by-design ad model, to protect user privacy by prevention rather than cure.

In the first part of this thesis, we study several vulnerabilities in current advertising systems that leak user data from advertising companies to external entities. First, since targeted ads are personalized to each user, we present an attack exploiting these ads on the fly to infer user private information that have been used to select ads. Second, we investigate common ad exchange protocols, which allow companies to cooperate in serving ads to users, and show that advertising companies are leaking user private information, such as web browsing history, to multiple parties participating in the protocols. These web browsing histories are given to these entities at surprisingly low prices, reflecting the fact that user privacy is extremely underestimated by the advertising industry.

In the second part of the thesis, we propose a privacy-by-design targeted advertising model which allows personalizing ads to users without the necessity of tracking. This model is specifically aimed for the two newly emerging ad technologies - retargeting advertising and ad exchange. We show that this model provides strong protection for user privacy while still ensuring ad targeting performance and being practically deployable.

I dedicate this thesis to my wife, Minh Trang,
and my little son, Minh Anh.

Acknowledgments

First and foremost, I would like to thank my family for their unconditional love and support. They have been always beside me to share and to help me overcome the most difficult moments in this journey.

Second, I would like to express my sincere appreciation to my advisor Claude Castelluccia, who gave me the opportunity to work in this interesting topic, for his great guidance and help during my PhD. I would also like to thank my co-advisor, Mohamed-Ali Kaafar for his help and advice, especially at the beginning of my PhD life. I would like to thank Gergely Acs for the invaluable discussions we had which helped me significantly advance in my work.

I am very thankful to all my colleagues at INRIA in general, and in Privatics team in particular, for their collaboration and for making my time at INRIA enjoyable. I extend many thanks also to all my friends for their support and the great moments we have shared in life.

I would like to express my gratitude to my doctoral committee for their helpful comments and discussions.

Finally, I gratefully acknowledge the financial support of my studies by the French Ministry of National Education.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In targeted advertising, companies track users' online activities with the aim to personalize ads. Even though this practice brings economic benefits, it raises significant concerns about potential privacy violations. This thesis aims to help companies and privacy advocates find ways to enhance user privacy in targeted advertising. Firstly, it exposes some vulnerabilities in current ad systems which potentially leak user private data from advertising companies to external parties, resulting in their loss of control over this data. Secondly, it proposes a technical design of a privacy-preserving targeted advertising system which provides strong protection for user privacy while retaining the current business and system models.

In this first chapter, we highlight main privacy challenges in targeted advertising which motivate our work. We also give background information about trackers and their roles in the advertising system. We then describe all possible privacy leaks related to these trackers, with the aim to position the threats that we expose. Subsequently, we analyze major privacy-enhancing initiatives proposed to date and based on which explain our choice of privacy-preserving targeted advertising. Finally, we summarize our contributions and present the thesis organization.

1.1 Privacy Challenges in Targeted Advertising

Online advertising brings substantial revenue for Internet companies and is the key to the development of free content and services on the Internet. Consequently, increasingly sophisticated methods have been developed to improve the efficiency of advertising.

Online *targeted advertising* (or *behavioral advertising*, or *interest-based advertising*) involves tracking Internet users' online activities (e.g., reading habit, search or purchases) across websites and over time, inferring their interests and characteristics (e.g., age or gender), and personalizing ads to them based on these data. Since introduced in 1990s [2][3], targeted advertising has been quickly evolving and actually accounts for a significant part of online advertising. The practice has been acclaimed by the industry as an efficient

marketing tool [4][5][6].

Theoretically, the benefits of targeted advertising are manifold. Personalized ads likely increase ad views and users' purchases or engagement, which consequently increases advertisers' revenue. In addition, publishers (i.e., content providers which sell ad spaces on their websites to advertisers) also get higher revenue as advertisers tend to pay more for ads which yield higher performance. In return, higher revenue motivates publishers to provide better (free) content or services to users. In can be argued that users are also beneficial from useful and relevant ads which fit their interests.

Unfortunately, targeted advertising is often coupled with privacy concerns as advertising companies increasingly track and analyze user activities, which might contain very sensitive information. These data, if misused by trackers or leaked to other entities (e.g., the government), may cause harms to users physically or mentally. Possible harms include embarrassment, price and service discrimination, or termination of employment.

On the other hand, privacy concerns bring consequences to the advertising industry. They erode user trusts on Internet companies and, as a result, users may turn to use anti-tracking or anti-advertising tools as solution, which is negative for the advertising business. Additionally, despite tremendous amount of time and effort spent in dealing with privacy watchdogs and lobbying privacy policy makers, the advertising industry still suffers from adverse regulations and law enforcement actions against their tracking behaviors [7][8][9]. Given exploded privacy concerns in recent years, the sustainability of targeted advertising is only possible if this practice conforms to privacy requirements.

The dilemma is that any attempt to enhance user privacy by limiting user data from being collected by companies may also limit targeting performance and consequently reduce advertising revenue, which is unwanted from the economic perspective. Addressing privacy problems is therefore posing multidisciplinary challenges from both technical and economic points of view. In the following, we highlight two main technical challenges:

1. **Detecting privacy leaks in targeted advertising:** Laws and regulations increasingly empower privacy watchdogs with audit and law enforcement rights to prevent trackers from abusing user sensitive data. Unfortunately, data hold by these trackers may intentionally or accidentally be leaked to other entities through their business practices. The data leakage may significantly enlarge the risk window to user privacy and obstruct the enforcement of accountability. Since ad technologies are evolving quickly and sophisticatedly, there needs to be constant effort to study and remedy privacy flaws in these technologies.

2. **Privacy-preserving targeted advertising:** Since detecting (and fixing) privacy problems in targeted advertising is necessary, it can only be considered a short-term approach. Enhancing user privacy needs a privacy-by-design long-term solution to

protect user privacy by prevention rather than cure. A possible direction is to design a targeted advertising system which does not rely on tracking.

1.2 Identifying Trackers

The core entities of an advertising system are *advertisers* and *publishers*. Advertisers, e.g. hotels.com, want to promote their products to users by showing ads to them on the websites they visit. Publishers, e.g. nytimes.com, develop websites providing content or services to users and sell ad spaces on their sites to advertisers.

Since advertising has been evolving with millions of publishers and advertisers, it is too difficult for them to buy or sell ads directly with each other. Consequently, there are many other companies that play intermediary roles between advertisers and publishers. The two most common types of these companies are ad network and ad exchange.

– **Ad Network:** An ad network, e.g. Google Adsense, is a company that collects ads from advertisers in order to display to users on publishers' pages. The role of ad network is to maximize the advertising efficiency for advertisers and advertising revenue for publishers.

  In targeted advertising, ad networks track users across websites and build user profiles. When a user visits a publisher page, the ad network selects ads that best fit the user profile to display to the user. For example, a Google's user profile might contain user's interests, location, age and gender; a user who is interested in sport is more likely to receive sport-related ads than others.

– **Ad Exchange:** The Ad Exchange (ADX) is a company that allows ad spaces on publishers' pages to be traded in auctions at real time. When a user visits a publisher's page, the information about the ad space and the user are made available to a number of registered ad buyers. These buyers determine at real time whether they want to take the ad spaces, at what prices, and with what message. These information go through an auction, and the winner has the right to serve ads to the user.

  In the ad exchange model, there are two other entities that help publishers and advertisers in participating in the auctions: Demand-Side Platform (DSP) to represent advertisers, and Supply-Side Platform (SSP) to represent publishers. The goals of these entities are to help their customers in optimizing their strategies: DSPs help advertisers buy the most suitable ad spaces at reasonable prices, while SSPs ensure that publishers's ad spaces are sold at most competitive prices.

  An ad network can play either role, DSP or SSP, in an ad exchange depending on their remnant ad or ad space inventory to make available through ad exchange.

Besides, a SSP can also play the role of an Ad Exchange: they manage the ad spaces of their customers (publishers) and make them available through their own exchange.

ADXs, DSPs and SSPs extensively track users and use their profiles in optimizing their strategies. For example, DSPs leverage user profiles to make their buying decisions in ad exchange, ADXs use user profiles to classify users and only send bid requests to appropriate buyers, SSPs use user profiles to determine the minimum selling prices at auction.

An important party in targeted advertising systems is the *data broker*, e.g. BlueKai. Data brokers collect user information from various source, online or offline, build user profiles and sell these profiles to interested entities, e.g. an ad network or ad exchange.

Terminology. In the scope of this thesis, we use the term *ad broker* for all entities that are operating in the middle between advertisers and publishers, be they Ad Network, Ad Exchange, SSP or DSP; these parties share the same set of characteristics or behaviors: they maintain user profiles by tracking or buying data from data brokers, and take part in the connection among publishers, advertisers and users.

Identifying trackers. Most privacy concerns are related to ad brokers and data brokers, which centrally collect user data across many websites on the Internet, often without user knowledge or consent. They are operated by companies which users might have never heard of, have no relationship with, and "would not choose to trust with their most private thoughts and reading habits" [10].

1.3 Privacy Leaks

In response to public concerns about user privacy, most trackers claim that they use the tracking data solely for advertising or improving their content and services. In its privacy policy [11], Google states that "We use the information we collect from all of our services to provide, maintain, protect and improve them ... to offer you tailored content – like giving you more relevant search results and ads.", and that "We will ask for your consent before using information for a purpose other than those that are set out in this Privacy Policy." In fact, these companies are subject to legal punishment if they violate these written policies, for example under the U.S. Federal Trade Commission Act which is against "unfair" or "deceptive" commercial practices [9].

However, even under the most optimistic assumption that these companies can be trusted of not abusing tracking data to cause harms to users, and that the law can regulate such malicious actions, there are various possibilities that the data maintained by these

companies can be intentionally or accidentally leaked to other entities, resulting in their loss of control over these data. In the following, we summarize the main types of these potential privacy leaks.

Security of data storage is not sufficient: When users' data are stored on the trackers' servers, they might be vulnerable to attacks on the storage if adequate security measures are not taken. For example, a recent intelligence leak shows that the U.S. National Security Agency, in its surveillance program, tapped in to the private links that connect Google, Yahoo and Microsoft data centers around the world and secretly collected user data flowing through these links; these data are transmitted in clear [12]. The related companies are also the most prevalent trackers on the Internet.

Trackers sell data to other entities: Ad brokers may not want to sell user data. As observed by Harper [13], "most websites and ad networks do not sell information about their users ... If an ad network sold personal and contact info, it would undercut its advertising business and its own profitability.". On contrary, the business of data brokers is to collect and sell user data to interested parties. These brokers have little power to control the use of these data once they have been sold to their customers. For example, data sold by Bluekai, a prominent data broker, to its customers are subject to the customers' data retention policies, which are different from Bluekai's [14]. The list of companies buying data from a data broker and their specific purposes are normally unknown to users [15].

Ad brokers accidentally expose user data through their advertising services: Ad brokers communicate with many other parties including advertisers (e.g., advertisers submit ads, specify user targeting criteria and view ad reports), other ad brokers (e.g. in the ad exchange model), and users (e.g. delivering ads). Since the communication data in these cases are often related to user personal information, they can potentially be exploited to infer user personal information. For example, advertisers can manipulate the user targeting criteria to target some user and use the ad report to learn information about the victim [16]. In the first part of this thesis, we study some vulnerabilities in current advertising systems that may accidentally leak user data from ad brokers to other entities.

1.4 Privacy-Enhancing Initiatives

As privacy concerns are widespread, multiple initiatives to enhance user privacy have been put in perspective. In this section, we discuss these initiatives and their limits, based on which explain our choice of a privacy-by-design approach.

1.4.1 Regulation and Self-Regulation

Privacy concerns have long drawn significant attention from legislative bodies [17][18]. For over a decade, the U.S. Federal Trade Commission (FTC) has been issuing multiple reports and urging for legislation to enhance user online privacy [19][20]. Although several acts have been proposed that give FTC rights to bring law enforcement against tracking companies, these rights are narrowly restricted to "unfair" and "deceptive" actions, i.e., when companies do not keep their promises to users. "As of May 1, 2011, the FTC has brought 32 legal actions against organizations that have violated consumers' privacy rights, or misled them by failing to maintain security for sensitive consumer information." [9].

With one of the core objectives being to "fend off adverse legislation and regulation", a group of advertising companies formed the Interactive Advertising Bureaus (IAB) [21], which then joined other similar organizations in a Digital Advertising Alliance (DAA). DAA has published a set of self-regulatory principles for its member companies with guidelines on enhancing user choices, knowledge and transparency. The two major DAA privacy initiatives are the advertising opt-out page [22] and the AdChoices icon [23]. The opt-out page allows users to opt-out from targeted advertising from companies participating in the program. The AdChoice icon is expected to displayed at a corner of targeted ads to notify users that the ads are tailored to their interests. By clicking on the icon, users can choose to opt out from targeted ads from the respective ad provider.

As for privacy advocates, the language of these self-regulations is "loose enough" and they contain exceptions to allow the concerning practices to sustain [24][25]. Similarly to the cases of other industry privacy self-regulation [26], there is little enforcement for those who do not follow the DAA's self principles [27]. In addition, the online opt-out tool, while challenging for users to understand and configure [28], merely allows users to stop receiving targeted ads; the tracking and the use of tracking data for other purposes are unaffected. The AdChoice icons are not recognizable by users or even rarely displayed [24].

In order to facilitate user choice of tracking, privacy advocates proposed Do-Not-Track (DNT). In DNT, users simply configure their web browsers to add a special HTTP header, *DNT:1*, in each request to notify trackers that they do not want to be tracked (*DNT:0* means users agree to be tracked); users do not have to visit a special website to turn it on as in the case of online opt-out tools. Companies which honor DNT must respect user choice expressed through DNT headers. To date, DNT has been implemented in most browsers, yet there have been merely 20 companies reportedly honoring the signal; no technical audit has been performed to check their compliance [29].

A W3C dedicated group was organized with the aim to standardize DNT [30]. However, the progress is slow and the group has not yet reached a consensus due to three contested aspects [31][32][33]:

- **Default setting:** While some believe that '1' should be the default value of DNT, the industry reject this idea, claiming that the choice should be explicitly made by users, and that DNT should be unset by default. However, an unset DNT header means that companies can still perform tracking, as in the case DNT is set to '0'.
- **Definition of DNT:** The industry wants to interpret the DNT signal as "do not target", i.e. advertising companies stop delivering targeted ads, while privacy advocates want it to be "do not collect information".
- **Architecture of negotiation:** How do sites get users who configure DNT:1 to opt in and to remain opted-in?

These conflicts are unlikely to be solved in the near future.

1.4.2 Blocking

Tracking requires that web browsers send user information in form of HTTP requests to trackers. Blocking tools identify the trackers' domains or the patterns of tracking requests and then block these requests from being sent to trackers. These tools are typically installed in form of browser extensions [34][35][36], or built-in features of web browsers [37]. The block list is often configured by the company providing the tool, or users can optionally select different lists from other third parties. Most of these tools offer users the ability to selectively block/unblock each tracker in the list.

Since blocking tools prevent the browsers from communicating with blocked sites, they can be fairly effective in preventing tracking. However, the solution is only realistic for advanced users, since these tools contain serious usability flaws, as discovered by a research [28]. These flaws are mostly due to inappropriate default settings to protect user privacy, lack of communication and feedbacks, confusing interfaces and negative affects on functioning features of websites. In addition, users cannot distinguish between trackers and therefore cannot make meaningful choice over which trackers to block. Finally, since the list of tracking companies and technologies are changing constantly, it is difficult and time consuming for tool providers to maintain block lists.

1.4.3 Privacy Preserving Targeted Advertising

Privacy initiatives presented above have been framed around the question of tracking or not tracking and how the choice should be given to users (e.g. by using opt-out tools, DNT or blocking tools). However, tracking puts user privacy at risk, while anti-tracking prevents targeted advertising and consequently undermines advertising revenue. Giving choices to users does not help avoid the zero-sum game between protecting privacy and enhancing economics: if privacy is enhanced, then advertising revenue is inevitably compromised. Not

surprisingly, the advertising industry has been reluctant to adopt these privacy enhancing initiatives, even though it has faced with increasing public and legal pressures to improve user privacy. On the other hand, reduction of advertising revenue impedes the growth of Internet; Internet users may suffer from paying higher prices or receiving content with lower quality.

Privacy Preserving Targeted Advertising (PPTA) initiatives aim to reconcile the two conceptually conflicting goals of privacy and economics by proposing technical system designs that allow targeted advertising while not relying on tracking users. There have been several research proposals in this direction [38][39][40][41]. Their common idea is to keep users' data at their devices and shift the ad-user matching process to users' terminals or trusted devices, with the aim to preserve the confidentiality of user data.

While other ongoing efforts are progressing slowly due to the fact that they do not address the crux of the problem, which is the conflict between privacy and benefits of targeted advertising, PPTA appears to be a promising approach. PPTA seeks to accommodate all privacy and targeting requirements in a positive-sum win-win manner, rather than following a zero-sum approach which likely requires impractical trade-offs. Since privacy is integrated as a built-in feature in these systems, the major challenge of these approaches is to fit diversified demands and business models of targeted advertising without introducing significant additional cost (e.g. infrastructure and network cost) or imposing important latency in the operation of advertising systems.

1.5 Contributions

The potential privacy leaks and privacy-enhancing initiatives given above draw a landscape of privacy challenges in targeted advertising. We now explain where our contributions stand in reaching the goal of enhancing user privacy in targeted advertising.

Our first contribution is to show some privacy leaks in the current advertising systems that potentially expose user privacy from ad brokers to other entities, which fall into the third category of leakage described in Section 1.3. However, different from the work of Korolova [16], which investigated the leaks of user personal information when ad brokers interact with advertisers, we study the information leaks when these entities interact with users (delivering targeted ads) and with other ad brokers in ad exchange.

We prove that, since targeted ads are tailored to each individual user, they can potentially expose user personal information. These ads, if not well protected in transmission, could be a potential source for any eavesdropper to collect and infer a lot of user information. In addition, we investigate common ad exchange protocols and show that companies do not actually take user privacy as priority in designing such protocols, as user privacy can be easily leaked to many parties taking part in the process.

Existent and potential privacy threats, including the privacy leaks presented in this thesis, motivate the need of a privacy-by-design approach. The second contribution of this thesis is to propose a technical design of a privacy-preserving targeted advertising system. This design is aimed to a newly emerged advertising technology, retargeting advertising (which aims to target users' exact intentions or online actions), and the ad exchange model. Retargeting and ad exchange protocols were not mentioned in previous works [38][40][39][41].

In summary, our contributions include:

1. **Privacy leak in targeted ads delivery:** We present a novel attack that exploits user personal information from ads. The attack consists of a filtering technique to filter targeted ads, and an inference technique to infer user private information. We show that any entity who has access to a small number of ads can retrieve a significant part of a user profile built by the ad network. The proposed attack presents an example of privacy risks related to fine-grained targeting technologies, which potentially happen not only in targeted advertising but also in any other personalized content and services.

2. **Privacy leak in ad distribution system:** We conduct a privacy analysis of common ad exchange protocols. The results show that these protocols are prevalent and enable the dissemination of user private information among various participating companies, some of them can significantly increase their tracking capabilities in comparison with their own tracking mechanism. In addition, we show that the related companies get access to these user private information at monetary prices surprisingly lower than what users might expect, showing the fact that user privacy is extremely underestimated by the advertising industry.

3. **A privacy-preserving retargeting ad system:** We propose a practical solution to enable retargeting ads without the necessity of tracking. In our approach, user data are stored locally, transparent to users and under their control. We distinguish our approach by using homomorphic encryption to protect not only user data from ad brokers, but also ad personalization algorithms of ad brokers from users. Moreover, our approach is novel as it is the first privacy-preserving solution to retargeting advertising and ad exchange.

1.6 Organization

The rest of the thesis is organized as follows.

- Chapter 2 provides a background knowledge about tracking and possible resulting privacy problems. This chapter highlights why online privacy is important.

- Chapter 3 summarizes previous work which are related to our work in this thesis.
- Part I, including Chapter 4 and 5, focuses on privacy leaks in current advertising systems which potentially expose user private data from advertising companies to external parties.
  - Chapter 4 presents an attack that exploit targeted ads on the fly to infer user private information. It provides quantitative results through experiments with real users in the case of Google advertising network.
  - Chapter 5 studies privacy leakages in ad exchange protocols and provides a quantification of these leakages. It then presents an evaluation of user privacy by analyzing how advertisers value user private data in ad exchange auctions.
- Part II, which contains Chapter 6, proposes a privacy-enhancing solution.
  - Chapter 6 presents the proposal of a retargeting system without tracking. This chapter describes the desire goals and security assumptions, presents the technical system design, gives a security analysis, and provides an evaluation of additional cost.
- Chapter 7 concludes the thesis and discusses possible extensions to our work and other future research directions.

Chapter 2

Background: Online Tracking and Privacy

In this chapter, we describe tracking in a technical perspective. We then elaborate what trackers can possibly collect from tracking users, and what information they can potentially infer. Trackers often argue that profiling data is anonymous and cannot be associated to any specific person. On the other hand, they promote arguments that ordinary people would have "nothing to hide" unless they are doing something wrong. We analyze these arguments and show that they either are incorrect or only focus on a narrow meaning of privacy. We also discuss why privacy is important and why people should care about their private lives on the Internet.

2.1 Tracking Technologies

From a technical perspective, it is useful to consider three parameters in a tracking action. First, an *identifier*, e.g. IP address or a unique number that trackers generate and assign to a user, a device or a web browser. The identifier helps trackers associate multiple pieces of information they receive to a user, a device or a web browser. Second, a piece of *behavioral information*, e.g. visited site's url, which is related to a user's online activities. Trackers use these behavioral information to build user profiles associated with an identifier. Third, a *collection technique* that trackers used to retrieve identifiers and behavioral information and send them to their servers.

A tracking action happens if a tracker, using a collection technique, collects an identifier along with one or some pieces of behavioral information. A typical example is that a tracker, e.g. *DoubleClick*, puts its JavaScript tracking code on a website, e.g. *hotels.com*, to retrieve user's cookie (i.e. a unique number generated for each user browser) and the website url, and then send these information to its server. By performing this action on many websites, the tracker has a list of visited urls assigned to the user cookie, and based

on which infer user information such as age, gender and interests.

In the following, we clarify these technical details of tracking by elaborating possible collection techniques, identifiers and behavioral information.

2.1.1 Collection Techniques

Web page owners embed their own or third-party tracking scripts on their websites. When a page is loaded to the user browser, the tracking scripts retrieve information related to the visit (Information retrieval) and send them to the remote server (Information transmission).

- **Information retrieval:** When a tracking request is sent to a tracker, some information about the user is integrated by default by the web browser in the request header, such as the browser information, the IP address and the url of the visited page. In case trackers need more information, such as the user's screen resolution, fonts or mouse behaviors, a common solution is to use JavaScript to retrieve such information and then encode them into tracking requests' urls or cookies[1].
- **Information transmission:** Retrieved information are encoded in HTTP request headers, request urls or cookies. They are generally sent to remote servers in form of tracking requests which are initialized by either AJAX technology (e.g. by using XMLHttpRequest in JavaScript) or HTML components that load remote resources (e.g. iframe, image, flash object, etc.). A tracking code can include these HTML components, JavaScript or both of them. JavaScript code can issue requests to trackers by using AJAX or by dynamically creating such HTML components.

To exemplify the process, we consider a typical scenario: The Javascript code in a tracking script dynamically generates an HTML code to display an image, e.g. *<img src='http://tracker.com[...]'>*, on the hosting page, e.g. *example.com*. This HTML *img* element loads the image from the url encoded in its *src* property, namely *http://tracker.com[...]*. Upon receiving this request, the tracker records the user's visit to the page *example.com*. It responses with an one-pixel image which is then displayed on the current page, but too small to be visible to the user.

2.1.2 Identifiers

2.1.2.1 IP Address

Each device once connected to the Internet is assigned with an IP address, and subsequently uses this address to communicate with other entities on the Internet, e.g. send and receive messages. IP address is by default included in any HTTP request, and naturally can be

[1] Cookie will be discussed in more details in the next section.

used by trackers to identify a device in tracking. IP tracking, however, is not largely used due to the following reasons.

First, the IP address in many cases is dynamic and is not always the same with a device. For example, an Internet Service Provider (ISP) may dynamically assign a different IP address each time a device connects to the Internet, or a user with the same laptop changes IP address when he moves from home to work. In addition, IP addresses might be shared across many members in a work or public place.

Second, the IP address is considered personal information [42], and therefore should be restricted from being collected. Even people would admit that an IP address does not always directly correlate to a given person, in a significant percentage of cases, it can be used to identify a household. Given information that users normally give to an ISP, such as name and address, IP address can be combined with these information to identify a user.

2.1.2.2 Pseudonymous Identity

To overcome legal and technical barriers of IP tracking, most trackers use pseudonymous identities (ids) to track users. These ids are randomly generated numbers which are unique per user device or web browser. They are sent by trackers to users at the first time they interact with these trackers, and stored at user device or web browser. Whenever users interact with these trackers, pseudonymous ids are attached in the tracking requests to help trackers identify them. In the following, we describe different possible storage mechanisms to store pseudonymous ids on the client side.

Cookies: The idea of using cookie was first introduced by Lou Montulli in 1994, when he was working for Netscape Communications. Cookie is a small piece of data sent from a website and stored at user browser when the user is browsing this website. A cookie normally contains a name/value pair (e.g. "id=123") along with the domain to which it belongs, its expiration date and other information. A cookie is included by default into any request to the domain from which it was originated. For example, a cookie whose domain is doubleclick.net will be attached to any request to this domain.

Actually, cookie is the most common method to encode user pseudonymous ids. These ids are typically encoded in the name/value pair contained in the cookie. For example, Doubleclick assigns a pseudonymous id for each user browser and encodes the id in its tracking cookie as "id=[browser id]". Typically, a tracking cookie expires in two years since the last communication with the user. A cookie is stored inside a browser and only accessible to JavaScript code from the same domain with the cookie.

Super cookies: The most well known form of super cookies is Flash cookies, or Flash Shared Objects (FSO), which are set by the flash plug-in commonly installed inside a

browser. When users load web pages containing flash components, these components can save flash cookies to user devices and then use them the same way tracking codes do with regular cookies. However, different from regular cookies, flash cookies are stored outside web browsers and can operate across browsers. Moreover, Flash cookies never expire, and, since not stored inside browsers, these cookies are not detectable by browser privacy mechanisms or most of privacy tools which are installed as browser extensions. Consequently, companies can rely on data stored in Flash cookies to re-spawn normal cookies when they are deleted by the users. Soltani et al. [43] showed that flash cookies were commonly used by popular websites, some of them stored the same user data in both regular cookies and flash cookies, with the aim to re-spawn cookies when necessary.

ETags: ETags, or entity tag, is part of the HTTP protocol which allows web cache validation in web browsers. Actually, ETags is an optional field in HTTP header which is used by content providers to store a version number for each version of a page's content. The user's web browser stores the ETags of the latest version it receives, and communicates the current ETags to the server when it wants to update content from the same address. The server compares the Etags in the client's request with that of the server-side latest version. If the client's version is outdated, it sends a new version along with the new ETags. Otherwise, the server sends a simple response to notify that the local version is up-to-date and can be used.

In order to use ETags for tracking users, trackers simply set all ETags related to its domain to the same value, which is the user's pseudonymous id. Since the ETags is included in every request, it can be used as a cookie containing tracking ids.

HTML5 Local Storage: HTML5 Local Storage is a client-side storage mechanism that allows website to store data in the local device. It can be used by trackers to store and retrieve user ids for tracking. Data stored in this storage is persistent as Flash cookie. However, different from Flash cookies (which requires Flash plug-in), HTML5 Local Storage does not require any plug-in and therefore can be used as a universal tracking mechanism.

Others: Many other types of storage, such as Silverlight isolated storage or HTML5 Session Storage, are available and can be used for storing tracking ids [27][44]. Kamkar [44] introduced Evercookie, a JavaScript API, which enables tracking ids to be written to and read from at least 10 storage places of different types (including all those mentioned above) in a user's device, with the possibility of re-spawning data from a storage to the others, making it difficult even for experts to delete the tracking ids [45].

2.1.2.3 Browser Fingerprints

Information about a browser or a device, for example user agent, language, fonts, screen resolution, operation system, list of plug-ins, etc., if taken together, can be used to uniquely or nearly uniquely identify a browser or a device. For example, Eckersley [46] shows that, based on these information, a browser is unique among at least 286,777 other browsers, and that, among his sample of around 470,000 browsers, 94.2% of those with Flash or Java are unique. This study also shows that even though fingerprinting information can be changed rapidly, using a simple heuristic is sufficient to associate the new with the old fingerprint, with 99.1% of accuracy.

Fingerprinting information can be obtained *passively*. For example, information about operating system, user agent and language are included by default (by the web browser) in every HTTP request header. Other information, such as time zone, display settings, installed fonts and installed plug-ins, can be *actively* retrieved by JavaScript code. Passive fingerprinting is problematic since it does not leave any trace in the client terminal, and therefore cannot be detected. Active fingerprinting, however, can be detected by examining JavaScript code for suspected behaviors, such as font-probing actions [47][48].

2.1.3 Behavioral Information

Web history: The primary piece of data being collected online is web url. When a user visits a website which contains a third-party tracking code and the code initializes a request to a third-party server, the currently visited web url is included by default in the *Referrer* field of the request's HTTP header. It can also be retrieved by JavaScript, and then processed and transmitted with the request. As proved by a study [49], DoubleClick, for example, can know on average 39% and, in a maximum case, up to 61% of a user web history. The Wall Street Journal found that, when user visits a popular website, dozens of third parties were aware of his visit [50].

User exact intention: For the purpose of retargeting[2], advertisers tend to track not only the websites users visited, but their exact intentions (e.g. to buy a specific product) on these websites. For example, if a user browses for a hotel on *hotels.com*, the id of the hotel will be transmitted to the tracker. In this case, the tracker cooperates with the website owner to retrieve the hotel providing this id, and therefore knows exactly what the user is interested in, e.g. a hotel in Barcelona, Spain. Even though this information in some cases can also be inferred from the url of the website using data mining techniques, retargeting trackers

[2]Retargeting is the technology used by advertisers to retarget, i.e. advertise, users with the exact products they previously showed interest in on their websites. Retargeting will be described in more details in Chapter 6

retrieve this information with absolute precision. In addition, they can track, for example, user purchase intents even in his private zone (e.g. pages which require logged-in).

Detailed behaviors: Many companies[3] advertise technologies to track user mouse movements, clicks, scrolls, keystrokes and form fills on web pages and construct detailed heat maps of these behaviors. Furthermore, a recent study [51] shows that mouse movements can be strongly correlated with eye movements. Mouse behaviors tracking, therefore, can be used to infer user attention (eye gaze) and build user attention flow patterns on web pages.

2.2 What They Know

The fact that users' activities on the Internet can expose a lot of information about them is not hypothetical or opaque, but real. Tracking data is publicly used by some ad brokers to infer user characteristics such as age, gender, used language and interests [52][53]. For example, Google maintains a set of around 1000 categories which include every aspect of life such as entertainment, shopping, law, health, finance, etc. Many of them are quite sensitive and detailed such as "/Finance/Credit & Lending/Loans" or "/People & Society/Family & Relationships/Family/Parenting/Adoption".

BlueKai can sort users into 30,000 highly detailed market segments like "light spenders" and "safety-net seniors", according to the New York Times [54]. In general, the company can estimate users' average net worth, political views, interests, spending habits, and more [54][55]. In the New York Times article [54], the author created two different profiles in two different browsers (running on the same computer) by browsing travel and shopping sites, searching for flights, cars, jewelries and visiting political websites and then observed the profiles created by BlueKai. What he found are surprisingly detailed and intrusive: One profile is "someone who makes between $\$60,000$ and $\$74,999$ a year, lives in Portland, Me., is interested in luxury cars, celebrities and TV, may have bought a cruise ticket, is an ideal candidate to take out a mortgage and a 'midscale thrift spender'.", while the other is "someone who lives in Los Angeles, Long Beach or Santa Ana, runs a large company with more than 5,001 employees and cares about advertising and marketing".

According to what publicly advertised to its customers in the BlueKai Audience Data Market [55], BlueKai possesses data about more than 160 million users who intend to buy a particular product or service in a near term, more than 103 millions users who are likely to interest in a topic or fall within a lifestyle category, and dozens of millions of users of other types such as business careers, past purchases, age, education level, household

[3]Examples include Mouseflow (mouseflow.com), ClickTale (www.clicktale.com) and Lucky Orange (www.luckyorange.com)

income or presence of children. In addition, the list of about 200 health-related Preference Data segments available on the BlueKai's Audience Data Marketplace can be found at [56]. Acxiom, one of the largest data brokers has, on average, 1,500 pieces of information on more than 200 million Americans [57].

By collecting products that users purchased, companies can characterize their purchasing habits and, by observing changing patterns of these habits, anticipate important events in their life, such as being pregnant, getting divorce or graduating [58].

In a recent report after investigating 9 major data brokers [20], the FTC reports that these entities are leveraging growing number of sources to compile evermore specialized customer segment, including those which are very sensitive, such as those related to ethnicity, income, religion, political leanings, age and health conditions. The segments found in the study can be quite specific and intrusive, such as "Urban Scramble", "Mobile Mixers", or health-related topics or conditions, such as pregnancy, diabetes and high cholesterol. The report also notes that these data brokers are frequently sharing data with each others and remarks the trend of associating online and offline data, such as between Datalogix and Facebook.

These details are only tip of an ice berg as they are what publicly published by trackers, excluding what they consider "sensitive information". In some specific cases, the collected data can be extremely more worrying. For example, Sparapani [57] found that "*A Connecticut data broker called "Statlistics" advertises lists of gay and lesbian adults and "Response Solutions" – people suffering from bipolar disorder.*", and "*"Paramount Lists" operates out of this building in Erie, Pa., and offers lists of people with alcohol, sexual and gambling addictions and people desperate to get out of debt.*", and that "*A Chicago company, "Exact Data," is brokering the names of people who had a sexually transmitted disease, as well as lists of people who have purchased adult material and sex toys.*"

In summary, as noted by the FTC, "The extent of consumer profiling today means that data brokers often know as much – or even more – about us than our family and friends, including our online and in-store purchases, our political and religious affiliations, our income and socioeconomic status, and more" [20].

2.3 Identifiability

Since trackers mostly use pseudonymous identifiers that are associated to a web browser, not a person, they often argue that tracking data is anonymous. However, there are quite a few possibilities that the data collected by these companies can technically be associated with real users. For example, users often provide their real identities to websites that they trust. These websites, however, can intentionally or unintentionally leak these information to third party trackers, so that these trackers can map users' pseudonymous identities with

their real identities. In addition, the data they collect can be de-anonymized given some prior knowledge about users which can be collected from public databases.

It is sufficient for trackers to link pseudonymous identity with a real identity just once to internally de-anonymize the user. Even if users manage to delete pseudonymous identity (e.g., by deleting cookies), the new pseudonymous identity assigned to them can be link with the old one in various ways, e.g. by observing the IP address. In the following, we elaborate various technical possibilities that tracking data can be associated to user identity, based on a taxonomy proposed by Narayanan [59].

2.3.1 Pseudonymous Identity Can Be Linked with Real Identity

The third party is also a first party: Many companies that are tracking users on the web, such as Google or Facebook, also have first-party relationship with users. For example, Facebook provides online social network services that require users to provide their real name in order to use these services. Moreover, Facebook put its widgets (e.g. "Like" button) across websites to track users. As the cookie sent from these widgets are also those sent to Facebook when users connect to Facebook as first party, Facebook can link these cookies with user identity such as their real names. Similar examples can be found in the case of Google, Yahoo, Twitter, and many others.

Leakage of identifiers from first party to third party: Several research studies [60][61][62] found that first parties sites, including social and non-social ones, are actually leaking user identifiable information, such as user social network id, username, email address or zip code, to first party sites. For example, some first-party websites include user name or email address in page urls or page titles when users log in. As this data is collected by trackers, users personal identifiable information could be retrieved by parsing the data. Since user cookies are included in tracking requests, it is trivial for trackers to map these cookies with user personal identifiable information.

A first party sells the user's identity : Some first party companies make a business of collecting user personal identities and sell them to third parties. For example, eBay was a prime data provider of matching services which match user offline data to their online cookies. Since eBay was a marketplace operating online, it assigned cookies to users. At the same time, it required users to register with their real names and email addresses before they can make orders on its marketplace. Consequently, with user cookies linked with their identities, eBay can provide services to other companies to sync users' offline data to online cookies. eBay shut down the service in 2011 [63], but plenty of other companies operating online which require user registration could have the same data, for example large e-commerce or hotel booking sites.

2.3.2 Data De-anonymization

User profiling data in a pseudonymous database can be de-anonymized by matching it with a public identified dataset. We illustrate this process through a simple example. Assuming that, in a certain day, a user visits websites belonging to his hometown newspaper, his university and his company, and that there is high probability that he is the only one who visited all these web sites at the same day. Consequently, anyone who knows his such visits can reasonably identify him in an anonymous web history containing these websites during that day. As a user web history was shown to be quite unique [64], such identification technique is likely feasible given an adequate amount of identified data is disclosed.

Since users increasingly participate and leave their public *footprints* on many websites, e.g. by putting likes and comments, these information can be collected and used to construct a public identified dataset about them. Even though users might use different usernames on different websites, in a significant proportion of the cases, usernames belonging to the same user have been shown to be strongly related and linkable [65].

The feasibility of data de-anonymization can be seen in a study [66] in which Narayanan and Shmatikov showed how to deanonymize the Netflix's published database, which contains anonymized movie ratings of 500,000 Netflix subscribers, by matching it with non-anonymous movie rating database on the Internet Movie Database (IMDb) website. The authors proved that anyone who knows a little bit about a subscriber can identify his record if it is present in the dataset. The conclusion is based on surprising statistics in the study: With 8 movie ratings and dates, 99% records can be uniquely identified in the dataset. For 68% of the record, even two ratings and dates are sufficient.

2.4 Why We Should Care about Online Privacy

2.4.1 Counter Arguments

Tracking is anonymous: The era that people are anonymous on the Internet has possibly gone. In the previous section, we showed that re-identification techniques are getting increasingly sophisticated, e.g. correlating users' online with offline data, and actually employed by certain companies. As users tend to engage more with the Internet in their daily lives and consequently leave more and more footprints in the online world, the possibility that they can be identified by companies are getting higher. Nowadays, it is safe for everyone not to expect that his identity is protected when browsing the Internet.

Nothing to Hide: In addition to the illusion of anonymity, a common argument which pervades privacy discussions is that ordinary people have "nothing to hide" and that if they commit in doing bad or illegal things, they do not have the right to keep them private. In

Britain, for example, millions of public surveillance cameras have been installed by the government in order to monitor its citizens. One of the program's slogan is "If you've got nothing to hide, you've got nothing to fear.". Companies also advance these argument to justify their activities. As stated by Eric Schmid, CEO of Google, "If you have something that you don't want anyone to know, maybe you shouldn't be doing it in the first place". Various forms of the nothing-to-hide argument appear on blogs, websites and forums. They are even supported by many users who declare that "As long as you're not doing anything wrong, you have nothing to worry about".

These arguments are considered by Schneier as "the most common distort against privacy advocates" [67]. The legal scholar Geoffrey Stone [68] refers to it as an "all-too-common refrain." Solove [69] refutes these argument saying that it only focuses on a narrow concept of privacy, viewing privacy as a form of concealment of bad things or secrecy. He argues that privacy violation does not necessarily mean exposing deepest secrets of users. Instead, that our data are collected, analyzed and used without our consent, or even knowledge, is a privacy problem in itself.

In fact, privacy is an inherent human right and a basic human need [67]. Everyone has a right to keep something, including the most awkward ones, private from others. Privacy, as articulated by Warren and Brandeis [70], is fundamentally the "right to be let alone". Ironically, by nothing-to-hide argument, trackers impose a default assumption that people have "nothing to hide", and require them to prove that privacy invasion causes serious problems in order to have the right to protect their privacy.

The problem is not just that someone has something that he might consider wrong, but that we all have something that someone, in some context, will consider wrong. For example, the fact that one's grandfather is Jewish (or Sunni, or a communist) is not particularly discreditable, but in other times or circles such information could get him in troubles with serious discrimination or even threaten his life.

Despite serious objections against nothing-to-hide argument, many people still think that privacy problems are only abstract concerns. In the next section, we show that privacy violation can potentially and actually cause negative tangible impacts on people's lives from multiple perspectives, rather than simply provoking feelings of unease.

2.4.2 Potential Risks of Privacy Violation

Solove [69] argues that "Privacy is often threatened not by a single egregious act but by the slow accretion of a series of relatively minor acts." Viseu et al. [71] explain that privacy is an abstract concept and that people only become concerned when their privacy has gone. They compared privacy loss with "global warming": people know that global warming is negative, but "the immediate gains of driving the car to work or putting on hairspray outweigh the often invisible losses of polluting the environment." In the following, we

discuss several such negative impacts of privacy problems, which are expectedly increased given more and more user data are made available.

2.4.2.1 The risks of abusing user data

Given various possibilities of privacy leaks discussed in Section 1.3, which are studied in more details in Chapter 4 and 5, user data is not only collected and used by trackers, but can also be leaked to various other entities, including the government or employers. Collected information, when abused by these entities, can potentially cause harms to users physically or mentally. Possible risks include government surveillance and control, service discrimination, price discrimination and bad effects on employment opportunities.

- **Government surveillance and control:** As government surveillance serves the purpose of detecting and fighting against crime and terrorism, it can also mis-classify innocent people as related to these behaviors. When people are suspected, for example of engaging in to a criminal act, they may get in trouble with deeper government penetration in their private lives, or even be subject to some prohibition, such as denying a fly. They do nothing wrong but the government may still cause harms to them.

  Moreover, many people are not comfortable with government scrutiny, not because of wrongdoings but for political reasons. Government surveillance therefore might impede the work of, for example, human rights workers or journalists, by preventing ordinary people from cooperating with them. In general, such surveillance could, for instance, inhibit such lawful activities as free speech, free association, and "other rights essential for democracy" [69].

- **Service discrimination:** Tracking and profiling people might reveal that they have or potentially have a certain diseases. A person might be denied insurance by an insurance company knowing this information. Generally, user profiles can be used not only for marketing purposes but also, for example, for “risk assessment” by financial companies or insurers. For instance, a category like “Biker Enthusiasts” could be used to offer discounts on motorcycles to a consumer, but could also be used by an insurance provider as a sign of risky behavior [20].

- **Price discrimination:** Sellers and marketers have long discovered that differential pricing is more beneficial to them than distributing the same prices to all users. The barriers of doing that - including computation resource constraint and lack of user data - have increasingly been overcome by advanced technologies. Increasingly, offered prices can be significantly different to users based on what they have done in the past. For example, sellers tend to estimate for each user his willingness to

pay and the possibility that he switches to other providers and accordingly offer a suitable price [72].

- **Bad effects on employment opportunities:** Many cases have been noticed in which people got fired as their employers were not content with what they had posted on social media pages [73], even when these posts were set private [74]. Besides these widely known cases, there are various possibilities that revelation of users' private information or behaviors, such as looking for a new job, can cause harms to them. Even when they do not directly lead to a job loss, such disclosure may impede their promotion opportunities or cause discrimination. In addition, employers are having a trend to collect information about job candidates before job interviews. Exposing private information may lead to discrimination among candidates, for example by their religion, ethnicity, political views, medical conditions and so on.

A common problem in all these practices is that what companies or governments know about users might be partial, which bring distort pictures of them. Their processing results therefore can be incorrect, leading to incorrect decisions. An innocent person might be suspected by the government, a customer might be charged higher than the others even though he is not wealthier than the others, or a person might be denied of recruitment for some incorrect knowledge about him. The trouble is, when users fall in these cases, they often do not know based on which data these decisions were made or where the data come from, and therefore have no chance to correct errors or complain. Various possible harms might come from errors, non-transparency and unaccountability.

2.4.2.2 The risks of personalization

Companies are using tracking data in personalizing not only ads but also various other content and services to users. For example, search engines show people search results relevant to their previous queries and clicks, Facebook provides user's news feeds based on what they liked or commented, or Netflix recommends films to users based on those they have watched and their ratings. The idea of relevance can be illustrated by a quote from Mark Zuckeberg: "A squirrel dying in your front yard may be more relevant to your interests right now than people dying in Africa." [75] By analyzing user data, companies are increasingly providing users with what they think relevant to users, in expecting to increase users' engagement and ad views.

These personalization services can be problematic if they only provide users with content which are based on their established views and regular reading habits, instead of those that could challenge and broaden their knowledge. Since personalization is an increased trend, users might get trapped in a narrowly restricted online universe, or in other words, a *filter bubble* [76][77], where what they clicked on in the past determine what

they see next. As Pariser puts it, "you can get stuck in a static, ever-narrowing version of yourself — an endless you-loop."

2.5 Conclusion

Tracking and related privacy risks might be vague for many people. In this chapter, we gave a concrete and comprehensive view on these practices. We detailed how tracking takes place and what trackers can possibly know about Internet users. We then analyzed possible resulting consequences to people if they lose their online privacy.

Actually, an understanding that tracking is problematic can serve many purposes. First, this helps users build a right attitude towards tracking and therefore consciously take measures to protect their privacy and actively contribute to the debates around this topic. Second, this urges companies to put more attention to user privacy in their business and to adopt privacy-enhancing techniques whenever possible, instead of leveraging pretexts that there are technically no privacy problem. Third, it put pressures on legislative bodies to work out legislations to regulate tracking behaviors.

Although not discussed in this section, we acknowledge that tracking does bring benefits. For example, government surveillance to fight against terrorists is important and necessary, collecting users' behaviors might help serving them more appropriate and useful content, or revenue from targeted advertising could foster innovation and free content on the Internet. Our point is that, instead of arguing that privacy problems do not exist (e.g. tracking is anonymous, nothing to hide, etc.), a more acceptable argument should be that they do exist, but the benefits from tracking might, in short term and in some respects, outweigh some privacy loss. Only when the consensus about harms and benefits of tracking is established, that the process towards enhancing online privacy can be accelerated.

One possible way for companies to reduce privacy concerns is to enhance transparency in their data collection and use. Even though tracking might serve useful purposes, users need to know exactly what are collected about them, what information companies infer from collected data, and how they use these data in their business practices. Furthermore, users should be given choices to correct errors or remove sensitive pieces of information that they are not willing to share. In addition, data resulting from tracking should be open to external technical audits to check their conformity to privacy protection standards (e.g. differential privacy, k-anonymity and l-diversity). As a matter of fact, privacy concerns could be significantly scaled down if tracking companies show that they actually have "nothing to hide".

Chapter 3

Related Work

Given significant concerns about privacy problems in targeted advertising, there have been quite a few research work studying this topic from different perspectives. Naturally, many researchers try to *track the trackers* and provide an understanding of their tracking technologies and the prevalence of their tracking behaviors. Simultaneously, scores of tracking protection techniques have been developed with the aim to provide users with more transparency about tracking and options to prevent unwanted tracking practices. Some privacy researchers focus on existent or potential privacy leakage resulting from the use of user private data, urging related companies to put privacy as a priority in their business.

On the other hand, multiple solutions to strike a balance between protecting user privacy and enhancing economics of targeting have been put in perspective. From the technical approach, several technical designs have been proposed to enable targeting services without tracking users. From the economics angle, some privacy researchers tried to evaluate the monetary value of user privacy while some others proposed market mechanisms where user privacy can be traded and used as a commodity, providing monetary compensation to users who allow their personal data to be used by companies.

In this chapter, we summarize these previous work with the aim to provide a comprehensive view on the state of the art in the field and position our contributions.

3.1 Tracking the Trackers

Knowledge about trackers' behaviors can enhance users' awareness about how and to which extent their data are collected and, at the same time, support policy makers in making meaningful legislative decisions. Some comprehensive surveys about online tracking and privacy can be found in [27][78].

3.1.1 Prevalence of Tracking

Krishnamurthy et al. [79] found that increasing aggregation of data have been performed by a steadily decreasing number of entities over a long period. According to results from longitudinal experiments in this study, the penetration of the top-10 third-party servers across a large set of popular Web sites had grown from 40% in October 2005 to 70% in September 2008. Roesner et al. [49] showed that several trackers can capture more than 20% of a user's browsing behavior. Specifically, they point out that DoubleClick can track on average 39% and maximum of 66% of the pages visited by a user. Besides, Facebook and Google can track users across an average of 23% and 21% of a user web browsing history (respectively 45% and 61% in the maximum cases).

The Wall Street Journal (WSJ) analyzed 50 most popular U.S. websites and built their "exposure index" - to determine the extent to which each site let trackers track their visitors. The results showed that many of them expose user visit to at least 50 trackers. The highest number was recorded in the case of dictionary.com, with 234 trackers [50].

Online social network tracking significantly raises concerns, as these companies often have first-party relationships with users and therefore can easily correlate tracking data with user real identities (Section 2.3). As shown in [80], online social networks tracking are presented in up to 22% of top Alexa sites, the tracking is in form of social network embedded contents such as Facebook "Like", Google+ "Share" or Twitter "Re-Tweet" button. The authors emphasize that these tracking do not require users to log in to their social network accounts or take actions (e.g. click on the buttons), but affect even users who do not have accounts. Another WSJ examination of nearly 1,000 top websites revealed that 75% of them included code from social networks, such as Facebook's "Like" or Twitter's "Tweet" buttons [81].

Another measurement study of web tracking including the prevalence of such tracking items as cookies and webbugs was presented in [82].

3.1.2 Tracking Techniques

Roesner et al. [49] proposed a taxonomy to classify trackers by the different techniques used to track users. The authors proposed 5 types of tracking behaviors depending on how trackers interact with user browsers: (1) Third-party trackers, e.g. analytics services, track users within sites (e.g. tracking identifiers are different with different sites); (2) Third-party trackers use the same identifiers in third-party storages (e.g. cookies) to track users across sites; (3) Cross-site third-party trackers force users to visit their domains directly (e.g. popup, redirect) and therefore put them in position of first-party entities; (4) Third-party trackers rely on other trackers to leak tracking identifers to them in order to track users across sites; (5) Cross-site trackers have their own websites or services which users may

visit or use directly.

In addition, browser security principles can be breached by trackers with the aim to collect user information. [83] exposed the privacy-violating information flows in JavaScript applications that help third parties steal user cookies, sniff user browsing history and collect user behaviors such as mouse clicks and movements. [84] show that the "same-origin" principle, which prevent web components from different domains to interact with each other, is not guaranteed, and that companies can violate this principle and leverage the caching information to share persistent identifiers and snoop on a visitor's activities at other sites. Moreover, by observing visited link differentiation on browsers, third parties can query user history database.

Tracking by regular cookies is widely known and publicly used by trackers, and therefore can easily be prevented even with browsers' default mechanism (such as block or frequently delete third party cookies). Tracking by supercookies or other permanent storage mechanism is much more worrying since the data in these cases is permanent and hard to detect. Soltani et al. [43] found that Flash cookies are a popular mechanism for storing data on the top 100 sites ranked by QuantCast. Some of them were using Flash cookies to recreate HTTP cookies deleted by users. A follow-up study investigating tracking techniques using Etags and HTML5 Storage mechanisms appeared in [85].

In addition, fingerprinting is a new trend of tracking and has attracted many research work. Eckerley [46] studied the uniqueness of web browsers that might allow trackers to fingerprint users by their browsers. The author observed that, among browsers that support Flash or Java, an average browser carried at least 18.8 bits of identifying information, and 94.2% of them were unique in the studied samples. In addition, [86] uncovered that the use of HTTP user-agent string alone, or in combination with IP prefix can accurately identify up to 80% client hosts, and that the technique can be effectively used to link user cookies from different sessions even if the users manage to clear cookies (e.g. by using private browsing mode) (up to 88% of user cookies). [48] revealed the techniques that are used by commercial fingerprinting service providers. [47] quantified the prevalence of fingerprinting actions on the Web by examining suspected font-probing actions.

3.2 Tracking Protection Techniques

A large set of privacy-enhancing tools [35][34][87][36] have been developed to provide users with transparency and control over tracking. Most commonly, these tools apply block lists which contain the patterns of tracking urls to be blocked. The block lists are built from in-door experiments and/or feedbacks from the community. In some tools, users can choose a block list from any third party provider, and flexibly block/unblock each tracker in the list.

Since blocking prevents all communications with trackers, this may also stop useful functionalities. For example, Facebook embeds "Like" buttons on many websites to allow users to express their preference to the website's contents. However, the requests sent from these buttons (even when users do not click) are perceived as tracking requests since they notify Facebook about the websites users visited. ShareMeNot [49] is designed to prevent this kind of tracking while still allowing the functionality of the embedded content. Its approach is to strip cookies in requests sent from these components but allow these cookies when users click on the button. The tool is applicable for a wide range or online social network embedded contents, such as "Retweet" button from Twitter or "+1" button from Google Plus.

There have been several other approaches to prevent tracking. For instance, NoScript [88] blocks Javascript, Flash and other plug-ins, only allowing those from trusted domains to be executed. BetterPrivacy [89] helps users view and delete Flash and other super-cookies in their browsers. Jackson et al. [84] develop a Firefox extension to enforce the same-origin policy in browsers to avoid information leakage between different domains. However, the authors argue that, even with perfect same-origin policy and third-party cookie blocking, cooperating sites can still share user information with each other unless web browsers do not maintain any user long-term states (e.g. long-term cookies).

Instead of blocking trackers, TrackMeNot [90] obfuscates user profiles by issuing randomized queries to search engines. Similarly, in addition to traditional blocking function, DoNotTrackMe [87] mask user personal information such as email, phone number, credit card number when users are required to provide them to websites they interact with. By using DoNotTrackMe, users can still receive emails or phone calls without having to provide their actual email addresses or phone numbers, and therefore can stop receiving them whenever they want.

Some browser extensions [91][92] are aimed to prevent fingerprinting by spoofing user-agent strings in HTTP requests. However, [48] shows that these extensions are not effective. Interestingly, they may be used as an additional fingerprinting feature. [47] also analyzed the vulnerabilities in fingerprinting-counter approaches such as Tor Browser [93] and Firegloves [94].

Given the popularity of tools for preventing tracking, Leon et al. [28] studied the most common ones and found that these tools contain serious usability flaws that make them challenging for normal users to understand and use. These flaws include inappropriate default settings to protect user privacy, lack of communication and feedbacks, confusing interfaces and negative affects on functioning features of websites.

3.3 Privacy Leakage

Leakage from first party to third party: First-party sites, with which users directly interact and provide their information in order to use their services, may leak these information to third party trackers. Krishnamurthy et al. [60] show that Online Social Networks (OSNs) and their applications are leaking user's PII such as user id, email address or zip code to third-party aggregators, and it is possible to link these PIIs with user actions both within OSN and non-OSN sites. In another study [61], Krishnamurthy et al. showed that non-social first-party sites also frequently leak user information to third parties aggregators. Concretely, 55% of studied sites directly leak piece of private information, with the result increasing to 75% if the leakage of site userids is included. In addition, sensitive search strings on a significant percentage of health-care and travel Web sites are often leaked to trackers.

The Wall Street Journal tested the top sites in the U.S., including those related to sensitive subjects, and detected that many of them share users' personal details, including email, name, username, age/birth year, zip code or others, to third-parties companies [62]. Interestingly, the Journal's website, wjs.com, was also leaking users' email, name and age to scores of third-party companies. In response, wsj.com says that most of these data are transmitted in error.

Leakage from ad brokers to advertisers: Ad brokers often build intermediary layers to allow advertisers to personalize ads to users without accessing to user data. However, Korolova [16] showed that, in the case of Facebook ad broker, this is not enough to protect user private data from being leaked to advertisers. Specifically, she presented a set of attacks that allow advertisers to take advantage of the micro-targeting capabilities offered by Facebook advertising platform to learn user private data.

Facebook ad systems allow advertisers to target users based on a set of user profile features such as age, gender, workplace, interests and location, whether they are set public or private by the users. An attacker can simply choose a set of public information which likely uniquely identify a user (e.g. living in France, working at Inria, studied at a university in Vietnam), then configure his ad to be shown only to users having such information in addition to a piece of private information he wants to guess (e.g. age from 25 to 35), and then observe the ad report. If this ad is viewed by one user, then the guessed information is likely correct (the victim's age is actually between 25 and 35). In case the victim clicks on the ad, the attacker can learn more information about him beyond his Facebook profile. For example, a click on a dating advertisement may reveal that the user is actively looking for a relationship.

Facebook tried to fix this problem by requiring that the number of users targeted for an ad be larger than a minimum number. However, Korolova argued that it is not adequate

as, for example, the attacker can create fake accounts to satisfy the minimum number of targeted users.

3.4 Privacy-Preserving Targeted Advertising

As early as in 2001, Juels [38] proposed the idea of storing user profiles locally and presented several technical schemes for ad brokers to distribute personalized ads to users without knowing their profiles. These schemes represent different trade-offs between privacy and resource cost, ranging from simple approaches such as ad servers distribute all ads to users for them to locally select those most relevant to their profiles, to much more sophisticated ones based on Personal Information Retrieval (PIR) and mix networks. In the latter schemes, ad requests are encrypted by users's public keys and then perturbed and aggregated by mix networks so that ad servers can deliver ads to users without knowing which ad is actually requested by a user. Juels only focused on ad delivery schemes and did not consider other features of an ad system such as click fraud defense, billing and accounting.

Saikat et al. proposed Privad [39], a complete privacy preserving targeted ads model. In their design, a client software builds the user profile locally, a *dealer* acts as an anonymizing proxy to mediate the communication between users and ad brokers, while the communication is protected from the dealer with the public keys of ad brokers. Privad uses a publish-subscribe mechanism for ad delivery: the ad broker transmits ads to users according to their subscribed coarse-grained information (e.g., generic interest category such as "sport"), the ads are cached at the user's device, and the local software selects ads that best match the user profile when encountering an ad box. Since the client in Privad is implemented as an untrusted black box, a reference monitor is needed to gauge the traffic between the client and the network. Ad auctions in Privad can be performed at client, at server or at a trusted third party [95].

Adnostic [40] similarly profiles users locally but uses a different approach for ad rendering. In Adnostic, the broker transmits a set of ads solely based on the page that a user is currently visiting, and the local software selects ads that best match the user profile to display to the user. Ad view reports are encrypted so that the ad broker does not know which ad is actually displayed to the user, while an homomorphic encryption scheme allows them to aggregate these reports and correctly charge advertisers. Since Adnostic does not require any trusted third party and allow ad brokers to see user visited pages as they do in today's systems, it provides a weaker privacy protection in comparison to Privad which aims to technically protect every piece of user information.

Obliviad [41] shares the idea of storing users' profiles at their devices, however shifts current algorithms of ad brokers into secure co-processors (SC). SCs are provided by a

trusted third party (e.g. IBM) and are tamper-resistant from any external entities including ad brokers. To request ads, the user sends his profile in form of keywords to the SC through a secure connection (e.g., TLS). The SC subsequently selects and serves ads from the broker's database using these keywords. Obliviad implements a PIR scheme over an ORAM structure to hide all the database access patterns of the SC from the broker. Each ad is associated with an encrypted token containing billing information which can be decrypted only by the SC. The ad broker collects these tokens when ads are displayed, and periodically sends these tokens to the SC and get back aggregated results which can be used to charge advertisers.

Hardt et al. [96] formalized a framework for personalized ad delivery taking into account three parameters - user privacy, communication complexity and ad relevance - and proposed optimization algorithms to optimize ad relevance given two constraints on user privacy and communication complexity. In addition, the authors developed a privacy-preserving protocol for aggregators (e.g. an ad broker) to gather user statistics (e.g. ad click through rates) in a differentially-private manner, which is efficient even when a large proportion of participants might dynamically change or become malicious.

RePriv [97] proposes a client-side framework for content providers to inject their *miners* (in form of embedded codes) to run on the user's terminal. These miners collect user information according to specific purposes of each provider, then customize the provided content accordingly. RePriv is aimed for content providers in general which are not necessarily related to advertising. However, this approach can be integrated in privacy-preserving targeted advertising schemes which opt to store user profiles locally, for example in building and maintaining local user profiles.

3.5 Economics of Privacy

3.5.1 Value of User Privacy

Understanding the value of privacy is of great important for companies, privacy advocates and policy makers in finding solutions to privacy problems.

In order to evaluate the cost of privacy, Hann et al. [98] considered four dimensions of privacy concerns - collection, error, secondary use and improper access - and showed that websites need to offer substantial monetary incentives to overcome these concerns. For example, the value of dis-allowance of secondary use of personal information is worth between $40 and $50.

Many studies estimated the value that users attach to their personal data through sealed bid auctions in which users sell their data to interested parties. For example, Danezis et al. [99] found that the bid which users made for their location data to be used by third party

companies is around the median of 10 pounds. The median bid increases to 20 pounds if users were informed that their data might be used for commercial purposes. A larger study (over 1200 people from five EU countries) [100] reaffirmed the median value between 10 and 20 pounds. Carrascal et al. [101] let users valuate their private data at real time by pop-up questions during their web browsing. Their result shows that, on average, users evaluate the price of the disclosure of their presence on a Web site to € 7.

In a different approach, Beresford et al. [102] studied people's unwillingness to pay for privacy. In their experiments, participants were given the choice to buy a CD from two stores which are identical except one requiring more sensitive personal data, such as monthly income or date of birth, than the other. With one euro discount, almost all participants chose to buy from the vendor requiring more sensitive data. Surprisingly, when the two vendors offer the same price, the buyers bought from both shops equally. In a quick survey after the experiments, most participants claimed that they have strong interest in their privacy protection.

The two common concepts to understand how users valuate their privacy are *Willingness To Pay - WTP* (the monetary amount users are willing to pay to protect their privacy) and *Willingness To Accept - WTA* (the compensation that users are willing to accept for their privacy loss). Krasnova et al. [103] found that, on average, a user would be ready to pay between € 14.14 and € 17.24 a year if the social network providers refrained from using his or her demographic information for personalized advertising. Another experiment [104] indicated that users would pay € 9.45 on average to migrate their Facebook profile information to Google Plus in case Facebook shuts down its social network service and deletes all data.

Acquisti et al. [105] discovered users' WTA and WTP through user choices of anonymous or traceable gift cards with different values, and suggested that the way privacy choices are framed may affect users' decisions. Gideon et al. [106] found that when privacy information is made readily available (e.g., displayed in correlation with websites on a search engine), users may be willing to pay a premium for increased privacy protection (e.g., buying products with higher prices from more privacy-friendly websites). Even though WTA and WTP are different for different types of personal data, there are wide gaps between WTA and WTP - WTA tends to be higher in all cases [107].

In fact, the value that users attach to their private information might vary depending on various conditions. For example, users tend to put low value on their personal information such as age and weight if they are among typical values or if they bring 'positive' images of the users (e.g. normal weight) [108]. [109] suggests the impact of demographic characteristics, for example, people with lower personal income are likely to be less concerned about privacy than those with higher income. In addition, users' valuation of privacy can be different by countries and genders (e.g., women are possibly more privacy

sensitive than men) [100].

Although most studies in literature estimated the value of user privacy from the user perspective, this approach has certain limits. First, users often lack enough information to make their privacy decisions, and even with sufficient information, they tend to trade off privacy for short-term benefits [109]. Moreover, the emotional aspect of privacy decisions makes it difficult to evaluate user privacy. For instance, users' answers to privacy surveys are significantly affected by the wording of the survey questions [110]. Moreover, due to the limited size of experimented samples, surveys conducted with different groups of users might lead to contradicted results. For example, [99] suggests that users who travel more often valuate their location data more than the others, while [100] found that travel frequency does not affect users' valuation of their location privacy.

The limits of these approaches can be complemented by studies from the advertiser perspective. Advertising companies actually make monetary profits from using user personal data, and therefore can have a practical view on the value of these data. The work presented in Financial Times [111] provided an analysis of industry pricing data from a range of sources in the US. The authors showed that general personal information, such as age, gender and location is worth a mere $0.0005. A person who is having a specific intent, e.g. buying a car, is likely worth more at about $0.0021. However, the used data source and methodology of the study are not published.

3.5.2 User Privacy as A Commodity

Instead of preventing trackers from collecting user data, some researchers have proposed ideas of a privacy market in which these companies pay users for the use of users' data. The argument in these proposals is that economic incentives will possibly increase users' adoption and engagement to these models.

Ideas about a privacy market at a high level were discussed in [112]. Later on, Ghosh et al. [113] proposed a formal framework for aggregators to buy private data from multiple users at auction. The authors proposed auction mechanisms in which sellers (users) are encouraged to report the true evaluation of their privacy and designed optimized algorithms for data analysts to reach their goals, for example maximize their accuracy goals at a limit budget or minimize their payment in achieving a given accuracy goal.

Riederer et al. [114] proposed the concept of "Transactional Privacy" in which users sell information about their browsing activities to aggregators on a personal information market. Each time a user visits a website, he chooses whether or not to sell information of this visit on the market through a trusted third party. Given the user's consent to sell his information, the trusted third party makes it available at auctions where aggregators can bid to gain access to these information. In this model, aggregators receive raw data (e.g. web page urls), and therefore can flexibly process these data according to their specific

purposes. The goal of this model is to keep user data under their controls, incentivize them to share data and provide maximum utility to data aggregators.

Part I

Privacy Leaks in Targeted Advertising

Chapter 4

Privacy Leaks in Targeted Ads Delivery

In this chapter, we show that targeted ads, which are often sent in clear, expose users' private data to any entity that has access to a small portion of these ads. More specifically, we show that an adversary who has access to a user's targeted ads can retrieve a large part of his interest profile. This constitutes a privacy breach because, as illustrated in Section 4.2, interest profiles often contain private and sensitive information.

Specifically, we describe an attack that allows any entity that has access to users' targeted ads to infer these users' interests recovering a significant part of their interest profiles. Our experiments with the Google Display Network [115] demonstrate that by analyzing a small number of targeted ads, an adversary can correctly infer users' Google interest categories with such high probability as 79% and retrieve as much as 58% of Google Ads profiles. The attack described in this chapter is practical and easy to perform, since it only requires the adversary to eavesdrop on a network for a short period of time and collect a limited number of served ads.

4.1 Motivation

This work was largely motivated by the Cory Doctorow's "Scroogled" short story that starts as follows [116]:

Greg landed at San Francisco International Airport at 8 p.m... The officer stared at his screen, tapping...

-"Tell me about your hobbies. Are you into model rocketry?"

-"What?"

-"Model rocketry."

-"No," Greg said, "No, I'm not."

-"You see, I ask because I see a heavy spike in ads for rocketry supplies showing up alongside your search results and Google mail."

-"You're looking at my searches and e-mail?"

-"Sir, calm down, please. No, I am not looking at your searches,... That would be unconstitutional. We see only the ads that show up when you read your mail and do your searching. I have a brochure explaining it ..."

The main goal of this chapter is to study whether such scenario would be possible today, and if one can infer a user's interests from his targeted ads. More specifically, we aim at quantifying how much of a user's interest profile is exposed by his targeted ads. However, as opposed to the above story, we do not consider ads that show up when a user reads his email or uses a search engine. These ads are often contextual, i.e. targeted to email contents or search queries. Instead, we consider targeted ads that are served on websites when a user is browsing the web.

The crux of the problem is that even if some websites use secure connections such as SSL (Secure Socket Layer), ads are almost always served in clear. For example, Google currently does not provide any option to serve ads with SSL[1] [117]. We acknowledge that in some scenarios the adversary can recover a user's profile directly from the websites he visits, i.e. without considering targeted ads. However, we show in this chapter that targeted ads can often improve the accuracy of recovered profiles and reduce the recovery time. Furthermore, in some circumstances, the victim has different browsing behaviors according to his environment. For example, a user at work mostly visits websites related to his professional activity, while he visits websites related to his personal interests at home. We show in this chapter that an adversary, such as an employer, that can eavesdrop on the victim's computer or network while at work can infer information about his "private" and personal interest profile. In other words, targeted ads constitute a covert channel that can leak private information.

Although there are various targeted advertising networks today, this work focuses on Google advertising system, which is "the most prevalent tracker" according to a survey of The *Wall Street Journal* [118]. However, our methodology is general enough to be extended to other ad networks. The problem of generality will be discussed in Section 3.1.

4.2 Targeted Advertising: The Case of Google

Google Display Network is a network of websites (also called publishers) that serves Google ads. Google receives ads from advertisers and then selects the appropriate publishers using various criteria such as relevant content, bid price and revenue.

In the Google targeted advertising model, Google Display Network sites are also used to track users as they browse the Internet. Each time a user visits a website that contains Google ads, i.e. a website that belongs to the Google Display Network, he sends his

[1] We verified this feature by browsing through several https websites (e.g. https://www.nytimes.com/).

DoubleClick[2] cookie to Google, along with information about the visited website. As a result, Google collects all the sites within the Google Display Network that have been visited by a user, and builds an interest profile from them. A Google profile is defined as a set of categories and sub-categories (see figure 4.1). For example, if a user visits a football site several times, Google may assign him the category *Sport*, or more specifically the subcategory *Sport* → *Football*. In addition, a Google profile may include location information and some demographic data such as the gender and age of the user. These profiles are then used to target ads to users.

A user can access and modify his Google Ads Preferences by accessing the webpage `http://www.google.com/ads/preferences` [119]. Furthermore, a user can choose to opt out of the Google targeted advertising if he no longer wants to receive targeted ads. Figure 4.1 displays an example of a Google user profile that contains potentially private

Figure 4.1: An Example of a Google Ads Preferences Page.

and sensitive information. For example, the “Job listing" category indicates that the user is probably looking for a job. A user might probably want to keep this information secret, in particular from his current employer. Furthermore, the category “Dating & Personals" indicates that the user is currently actively looking for a relationship, and the subcategories “Baby names" and “Adoption" that he has been recently visiting web sites related to baby adoption.

In its privacy policy, Google states that “We take appropriate security measures to protect against unauthorized access to or unauthorized alteration, disclosure or destruction of data.", and “We restrict access to personal information to Google employees, contractors

[2]In order to keep track of users visiting the Google Display Network, Google uses the DoubleClick cookie issued from the doubleclick.net domain which belongs to Google

and agents who need to know that information in order to process it on our behalf" [120]. Nevertheless, in this chapter we show that a portion of personal users' profiles could be leaked through targeted ads. Even if Google does not consider users' interests as "personal information", this data, which is related to users online activities, can be very private from a user's perspective.

4.3 Reconstructing User Profiles from Targeted Ads

As targeted ads are personalized to each user based on his profile, they can reveal a lot of information about users' interests. This section describes how an adversary who has access to an user's ads can derive part of his interests from them.

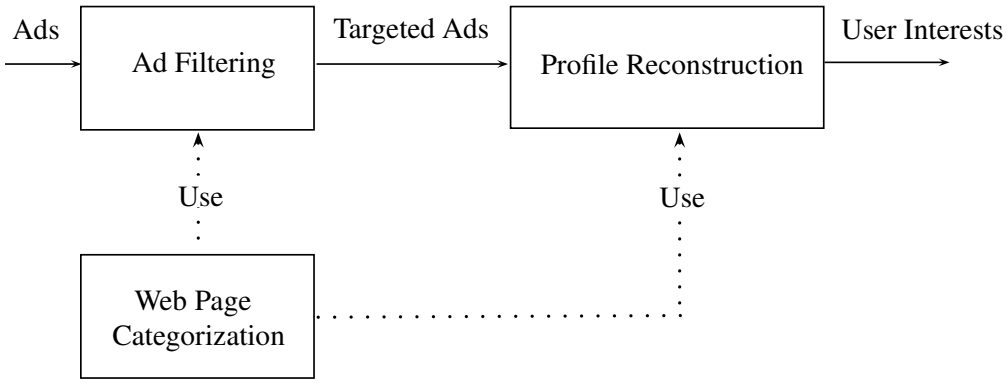


Figure 4.2: Filtering targeted ads and inferring user interests.

As shown in Figure 4.2, our approach is composed of two main phases. The first phase collects all ads served to a target user and filters them to only retain targeted ones. In the second phase, targeted ads are classified into categories and a profile is re-constructed. These two phases are detailed in the rest of this section. However, we first start by presenting two building blocks used by these both phases. The first one is an algorithm that is used to categorize webpages. The second one is a set of algorithms to compare categories.

4.3.1 Building Blocks

4.3.1.1 Web Page Categorization

The web page categorization tool is an algorithm that derives interest categories from a web page. This algorithm relies on the tool used by Google to generate Google Ads Preferences pages. As described in the previous section, each time a user visits a web page, Google derives some interests in the form of categories and updates the user's Google Ads Preferences page accordingly.

Ad Url	Categories
http://www.elasticsteel.net?ID=156	Beauty & Fitness → Fitness
http://www.livecarhire.com	Travel → Car Rental & Taxi Services
http://www.terracebeachresort.ca	Travel → Hotels & Accommodations Travel → Tourist Destinations → Beaches & Islands
http://www.sanibelbayfronthouse.com	Arts & Entertainment → Entertainment Industry → Film & TV Industry → Film & TV Production Real Estate → Timeshares & Vacation Properties Travel → Tourist Destinations → Zoos-Aquariums-Preserves
http://www.siestakeyaccommodation.com	Real Estate → Timeshares & Vacation Properties Travel

Table 4.1: Ad page categorization example

We use this tool to build our page categorization algorithm. Given a webpage W, our algorithm operates as follows:

1. W is visited and and the resulting DoubleClick cookie is saved.

2. A request is made to the Google Ads Preferences page with the previously saved cookie. Note that Google does not always update the Google Ads Preferences page upon a single web page visit. Usually, a webpage has to be visited multiple times to update the Google Ads Preferences page. Furthermore, we noticed that users' Ads preferences are updated after a period of time ranging between 1 and 2 minutes. Therefore, this step is repeated 5 times (heuristic value) every 2 minutes.

3. The Google Ads Preferences page is parsed and the corresponding categories are retrieved.

To evaluate the performance of our approach, we scraped 5000 ads from Google search page and 2000 sites from Google Display Network, classified them by the tool, and reviewed the results. We detected that almost all of these pages can be categorized by Google (more than 90%). We also manually reviewed the categorization results and observed that, although there are some irrelevant categories, the categorization generally reflects the content of each page. Table 4.1 presents several examples of ad page categorization.

It should be noted that relying on Google does not reduce the generality of our method. There exist many efficient techniques to categorize the content of web pages. For example [40] uses cosine similarity. This method is very efficient since it relies on social/crowd data (folksonomy) which is continuously updated, and is appropriate for fine-grained

categorization. We implemented this method and compared its performance with the Google-based categorization approach we described above. The obtained results were quite similar, with more than 60% of the categories overlapping. We therefore believe that our work can be extended to other ad networks, such as Yahoo! or Microsoft, either by applying their own categorization, or by simply using an existing webpages categorization technique. Note that Yahoo! and Microsoft also build users' behavior-based interest profiles and similarly to Google personalize ads to users according to their interests [121] [53].

4.3.1.2 Category Comparison Methods

Many of the filtering and evaluation algorithms presented in this chapter need to compare categories. We use three methods for this purpose: "Same category", "Same parent" and "Same root":

1. *Same category:* Two categories are considered equivalent in the "Same category" method if they match exactly.

2. *Same parent:* Two categories are considered equivalent in the "Same parent" method if they have the same parent category. For example, the two categories "Arts & Entertainment → Entertainment Industry → Film & TV Industry → Film & TV Awards" and "Arts & Entertainment → Entertainment Industry → Film & TV Industry → Film & TV Production" have the same parent category "Film & TV Industry", so they are considered equivalent to each other in the "Same parent" method.

3. *Same root:* Two categories with same root category are considered equivalent in the "Same root" method. For example, the two categories "Arts & Entertainment → Entertainment Industry → Recording Industry → Music Awards" and "Arts & Entertainment → Movies → Action & Adventure Films → Superhero Films" have the same root category "Arts & Entertainment" and therefore are equivalent to each other in the "Same root" method. Obviously, if two categories are equivalent in the "Same parent" method, they are also equivalent in the "Same root" method.

4.3.2 Extracting Targeted Ads

Ads provided by Google are either location-based, content-based (we call hereafter contextual, i.e. related to the visited page's content), generic, or profile-based (we call hereafter targeted, i.e. customized to users' profiles). In this chapter, we only consider targeted ads. We therefore need to filter out location-based, content-based and generic ads (see figure 4.3).

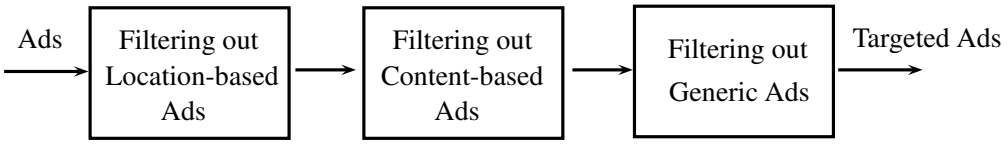


Figure 4.3: Filtering targeted ads.

Reconstructed profile
Beauty & Fitness → Fitness
Travel
Travel → Car Rental & Taxi Services
Travel → Hotels & Accommodations
Travel → Tourist Destinations → Beaches & Islandsl
Arts & Entertainment → Entertainment Industry → Film & TV Industry → Film & TV Production
Real Estate → Timeshares & Vacation Properties

Table 4.2: Profile reconstruction example

We conducted all experiments with users from the same location. As a result, the location-based filter is not used (and therefore not presented here). Furthermore, we consider that an ad is contextual if it shares at least one category with its displaying page (the page on which the ad is delivered). To filter out contextual ads, we therefore categorize, using the categorization technique described in Section 4.3.1.1, each ad and their displaying page. If at least one category is in common, the ad is classified as contextual. To filter generic (i.e. not customized) ads, we create a number of non-overlapping user profiles (i.e. profiles without any categories in common), and perform 10 requests to the tested pages[3]. Ads that are served independently of the requesting profile are then deemed generic and filtered out.

4.3.3 User-Profile Reconstruction

Given the targeted ads from the previous step, there are possibly many approaches to infer user information. In our work, we aim at reconstructing the Google-assigned interest categories which are presented as user profiles. In order to reconstruct a user profile, we categorize all of his targeted ads using our Google-based web page categorization tool. The reconstructed profile is then the set of resulting Google categories.

For example, considering the ads provided in table 4.1, the reconstructed profile will look as in table 4.2.

[3]The number of 10 requests is considered to be enough to get a sufficient ad collection while resisting well to the ad churn [122].

4.4 Evaluation

In this section, we evaluate the performance of our profile reconstructing technique.

4.4.1 Experiment Setup

Figure 4.4 illustrates the setup of our experiments. Our experiments are composed of two main phases:

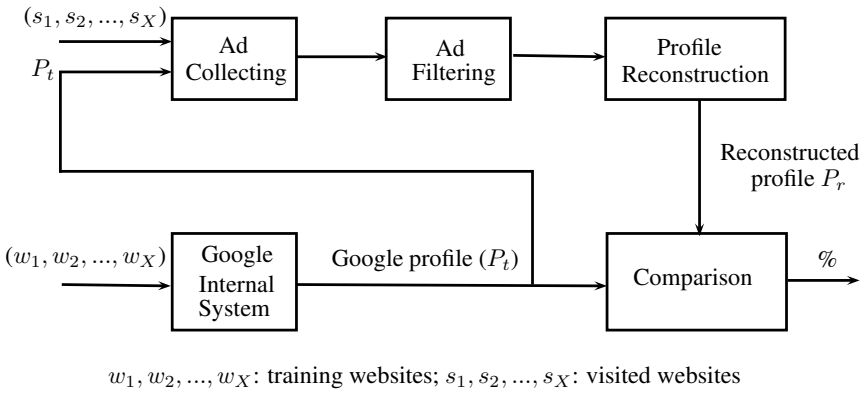


Figure 4.4: Experiment setup

Profile creation: In this phase, we create a set of profiles corresponding to different web users. Each of these profiles, that we call *targeted profiles,* P_t, is obtained by visiting several websites from a user's real web-history (i.e. list of websites that the user has visited). We refer to these websites as *training sites*. Each of them is visited 15 times to make sure it really affects profiles. We then retrieve the generated Google profile from the Google Ads Preferences page (this phase corresponds to the lower part of figure 4.4).

Profile re-construction: In this phase, we visit for each targeted profile (P_t) created as described above another set of websites, that we refer to hereafter as *visited websites*. As opposed to the training sites, each visited site is only visited once. The ads are then collected, filtered and the profile reconstructed as described in Section 4.3. We refer to the set of profiles we obtain as *reconstructed profiles*, P_r (this phase corresponds to the upper part of figure 4.4).

4.4.2 Evaluation Methodology

Dataset: Our target web-history data comes from a set of 40 volunteers who provided their list of websites they visited during two days. The first *X* websites in each profile were used as the set of training sites to create P_t. The *Y* following websites were used to build the reconstructed profiles, P_r, as shown in Figure 4.5.

In the presented experiments, X was set to 30 and different values of Y were used. The average number of root categories and categories in a targeted profile from X websites is displayed in Table 4.3.

	# of root categories	# of categories
X = 30	6.64	18.06

Table 4.3: Profile size statistics

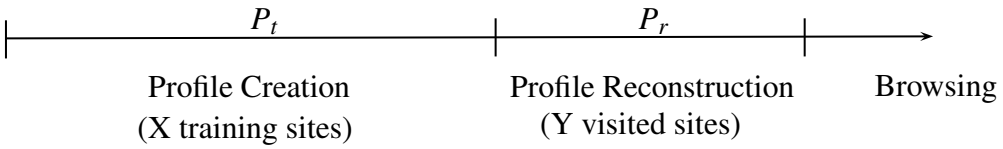


Figure 4.5: Profile creation and reconstruction.

Performance evaluation metrics: To evaluate the results, we compare each reconstructed profile with the corresponding original one. We compare profiles using the "same-category", "same-parent" and "same-root" methodologies described in Section 4.3.1.2. We evaluate the performance of our technique by computing the average *Precision*, *Recall* and *F-Measure* values of all reconstructed profiles. Precision is the fraction of rebuilt categories that are correct, while Recall is the fraction of original categories that are correctly rebuilt. F-Measure is the harmonic mean between Precision and Recall, defined as: $F = \frac{2.precision.recall}{precision+recall}$.

In other words, if we denote by $P_{r,c}$ the categories of the reconstructed profile P_r that are correct, and $P_{r,i}$ the categories of P_r that are incorrect, $Precision = \frac{|P_{r,c}|}{|P_r|} = \frac{|P_{r,c}|}{|P_{r,c}+P_{r,i}|}$ and $Recall = \frac{|P_{r,c}|}{|P_t|}$.

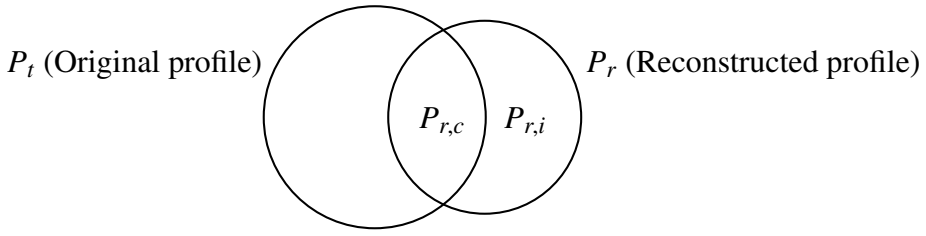


Figure 4.6: Illustration of Precision and Recall.

Adversary strategies: In order to evaluate the performance gain obtained by using targeted ads as opposed to only using visited websites, we consider the following three strategies:

- the adversary only uses visited websites ("Sites only").

- the adversary only uses targeted ads ("Ads only").

- the adversary uses visited websites and targeted ads ("Ads & Sites")

Tested scenarios: Finally, we consider two different scenarios, corresponding to two different browsing behaviors:

1. *HotSpot Scenario*: This scenario corresponds to the case where the victim is connecting to an open network and browses the Internet according to his interests. In this scenario, the X training sites and the Y visited sites are related to each others, i.e. generated from the same interest profiles. The goal of this scenario is to show that targeted ads can be used to boost the accuracy of profile reconstruction.

2. *Workplace Scenario*: This scenario corresponds to the case where the victim changes his browsing behavior during the reconstruction phase. In other words, profiles used to generate the training sites and the visited sites are significantly different. This situation happens, for example, when the victim is moving from his home to his work environment. The goal of this scenario is to study how much of the home profile leaks from the targeted ads shown at work.

In the following, we present, for the workplace scenario, how we select the visited websites so that they are largely separated from a user's interests. We first randomly select a set of Google root categories, namely "Autos & Vehicles", "Law & Government", "Books & Literature", "Beauty & Fitness", "Jobs & Education" and "Business & Industrial". We then get for each of these categories 500 websites using the Google Adwords Placement Tool [123]. This tool aims at helping advertisers to select websites to publish their ads. We then get for each user all of his root categories, and select a root category C that does not belong to them. The user's visited sites are then randomly selected from the 500 websites corresponding to category C. For example, if a profile contains 4 root categories: "Law & Government", "Books & Literature", "Beauty & Fitness", "Jobs & Education", then one of the remaining categories, "Autos & Vehicles" or "Business & Industrial", will be chosen for visited websites. We verified that none of our test profiles contains all the six visited categories.

Note that a website classified in a Google category according to Google Adwords Placement Tool may result in another category in Google Ads Preferences. For instance, Google may assign a website W to category "Arts & Entertainment". However, when categorizing this website using Google Ads Preferences, the result may include, in addition to "Arts & Entertainment", another root category, say "Books & Literature". Therefore, we cannot completely guarantee that the visited websites are totally separated from the training ones.

4.4.3 Result Analysis

Tables 4.4, 4.5, 4.6 and 4.7 represent the achieved Precision, Recall and F-Measure values in percentage with (X = 30, Y = 10) and (X = 30, Y = 15) for the hotspot and

workplace scenarios respectively. The rows in these tables specify the category comparison methods used to filter out contextual ads[4]. This comparison method is also used to evaluate the results (i.e. to compare the reconstructed profiles with the original ones)[5]. We remind the reader that these comparison methods are described in Section 4.3.1.2. The columns of the table specify the three different cases of profile reconstruction, using “Sites only", “Ads only" and “Ads & Sites", respectively. The tables show that the Ads-based information leakage is significantly high, with precision values ranging from 73 to 82% for reconstructed profiles evaluation based on recovering the root of categories solely from Ads. For example, in case ($X = 30$, $Y = 15$) in the workplace scenario, with “Ads only" and the “Same root” comparison method (used for both filtering and evaluation processes), we achieve Precision, Recall and F-Measure of more than 79%, 58% and 67% respectively (Table 4.7). The average number of targeted ads we observed accounts for approximately 30% of all collected ads in each case. We note that the results of the row “Same Category" show in general a relatively lower precision and recall values than the results of the “Same Parent" and “Same Root" rows.

	Av.# of targ. ads	Sites only Prec./Recall /F	Ads only Prec./Recall /F	Ads & Sites Prec./Recall /F
Same cat.	14.29	19.66/7.6 /10.96	18.04/7.06 /10.15	18.3/14 /15.86
Same parent	10.94	58.25/29 /38.72	53.67/19.38 /28.48	55.98/42.29 /48.18
Same root	9.24	79.26/51.44 /62.39	73.08/30.06 /42.6	79.6/68.33 /73.54

Table 4.4: Reconstructing Google profiles performance in Hotspot scenario ($X = 30$ and $Y = 10$)

	Av.# of targ. ads	Sites only Prec./Recall /F	Ads only Prec./Recall /F	Ads & Sites Prec./Recall /F
Same cat.	21.53	19.67/10.28 /13.50	15.71/8.47 /11.01	17.07/17.66 /17.36
Same parent	16.67	54.46/34.44 /42.2	51.26/23.54 /32.26	52.73/50.16 /51.41
Same root	14.4	75.57/61.13 /67.59	82.24/40.3 /54.09	78.5/80.52 /79.5

Table 4.5: Reconstructing Google profiles performance in Hotspot scenario ($X = 30$ and $Y = 15$)

Figures 4.7 and 4.8 display the variation of Precision, Recall and F-Measure when varying the number Y of visited web sites for each targeted profile, for different comparison methods. We observe that, for a given profile (i.e. when X and therefore $|P_t|$ are fixed), the

[4]For example, the row "same parent" displays results when ads are considered contextual if they share the same parent categories with the pages that display them.

[5]For example, the column “same parent” means that two categories are deemed identical if they share the same parent.

	Av.# of targ. ads	Sites only Prec./Recall /F	Ads only Prec./Recall /F	Ads & Sites Prec./Recall /F
Same cat.	19.23	2.92/1.05 /1.54	14.73/10.28 /12.11	11.84/10.94 /11.37
Same parent	13.6	9.09/3.99 /5.55	46.31/30.31 /36.64	34.39/31.56 /32.91
Same root	11.2	12.65/6.43 /8.53	78.07/53.96 /63.81	56.49/55.94 /56.21

Table 4.6: Reconstructing Google profiles performance in Workplace scenario ($X = 30$ and $Y = 10$)

	Av.# of targ. ads	Sites only Prec./Recall /F	Ads only Prec./Recall /F	Ads & Sites Prec./Recall /F
Same cat.	28.11	2.99/1.31 /1.82	13.44/11.95 /12.65	10.89/12.62 /11.69
Same parent	20.3	9.13/5.06 /6.51	44.95/33.8 /38.59	32.75/35.45 /34.05
Same root	17.13	14/8.61 /10.66	79.37/58.12 /67.10	55.85/60.1 /57.9

Table 4.7: Reconstructing Google profiles performance in Workplace scenario ($X = 30$ and $Y = 15$)

recall increases noticeably with Y, the number of visited web sites, while the precision is steady. This shows that the number of correctly reconstructed categories, i.e. $|P_{r,c}|$, increases with Y. This result is expected since when Y increases the number of collected ads also increases and as such the amount of available information is higher. However for a given X, the precision is not notably affected by Y, which means that the number of incorrectly reconstructed categories, i.e. $|P_{r,i}|$, also increases with Y.

In the hotspot scenario, the visited websites are largely relevant to the training websites, therefore reconstructing profiles from “Sites only" achieves the results as good as, if not better than, the results obtained from “Ads only" (see figure 4.7). However, when we combine both sites and ads in the reconstruction process, we get nearly the same Precision, while increasing Recalls remarkably (almost the sum of the two cases “Sites only" and “Ads only"). In this scenario, Ads are very useful to boost the performance since they allow the recovery of a larger part of the targeted profiles.

In the workplace scenario, the visited websites are considerably separated from training websites. Therefore, reconstructing profiles from “Sites only" leads to very poor results, whereas the “Ads only" technique achieves significantly better results (see figure 4.8). By combining sites and ads, we slightly increase the Recall while reducing the Precision. In this scenario, we observe that ads do indeed constitute a “hidden” channel that leaks private information about users.

While the performance of our method when evaluating the recovery of “root” and “parent” categories is notably high, we acknowledge that it can only recover a small proportion of a user’s actual categories (Precision varies between 10 and 18% when using the same category method for evaluation, and Recall ranges from 10 to 17%). We believe

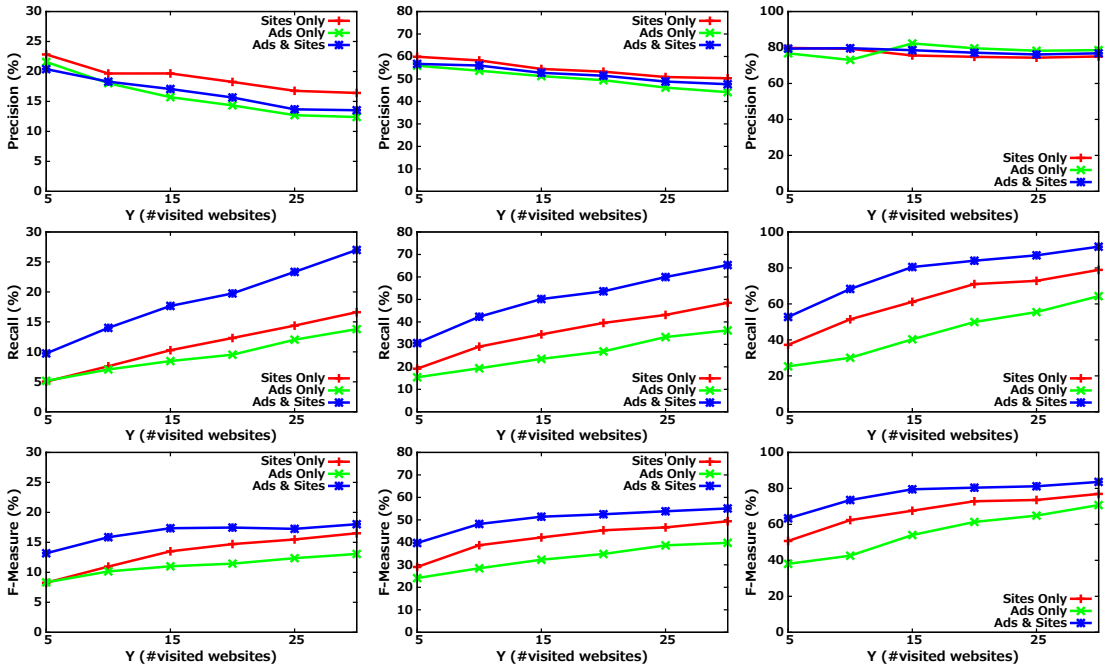


Figure 4.7: Precision, Recall and F-Measure with the “Same category", “Same parent" and “Same Root" comparison methods (from left to right respectively) used in both filtering and evaluation processes (**In hotspot scenario with** $X = 30$).

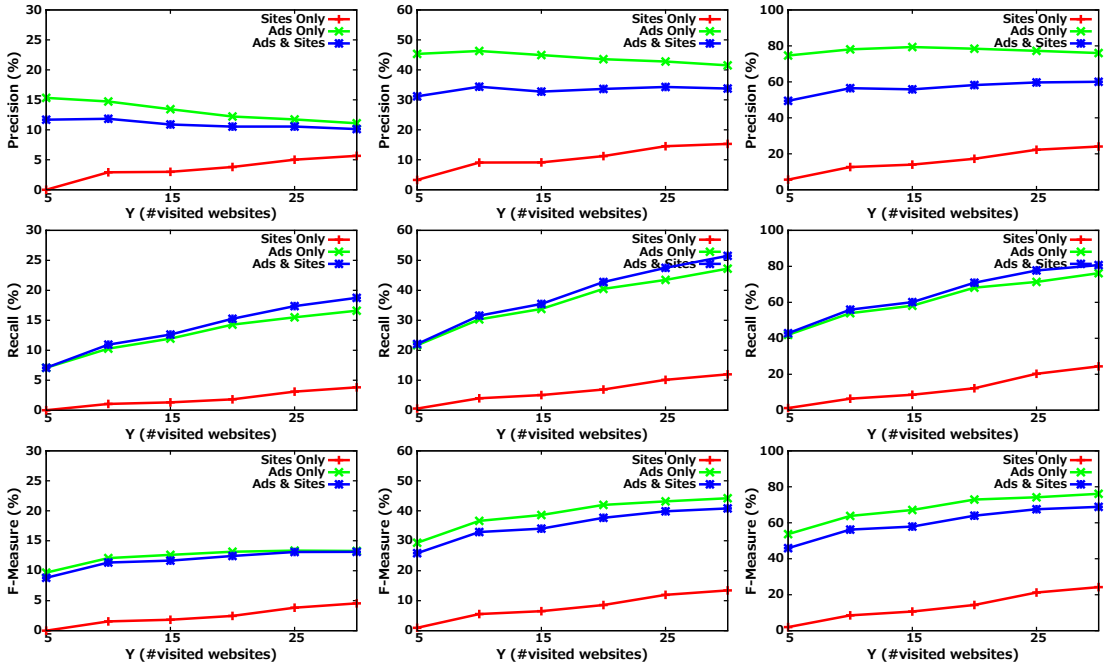


Figure 4.8: Precision, Recall and F-Measure with the “Same category", “Same parent" and “Same Root" comparison methods (from left to right respectively) used in both filtering and evaluation processes (**In workplace scenario with** $X = 30$).

there are several explanations for this result. First, Google might not be using the user’s actual categories to deliver ads, and instead might use the root or parent categories for a broader coverage of users’ interests. Furthermore, our ads classification method is probably

not optimal and could be improved in many ways. In particular, we took a simple and conservative approach in the filtering step by ignoring location-based ads. In addition, we did not consider remarketing ads that, as discussed in Section 4.5, may contain additional information about the users recent online activities.

However, even only 10 to 15% of an user's interest categories can constitute a severe privacy breach. To illustrate this statement, we ran our technique on the profile shown in Figure 4.1, and using targeted ads *only*, we recovered the profile shown in Figure 4.9. Among the recovered categories, the category "Online Communities → Dating & Personals" may constitute a private piece of information which a user might not be willing to share.

Figure 4.9: Reconstructed Profile.

4.5 Discussion

Countermeasures. In order to protect against this information leakage, the easiest solution today is to simply opt out of targeted advertising, frequently delete cookies or use ad-blocking software. Initiatives such as NAI (Network Advertising Initiative) [124], DNT (Do Not Track) [125] or TPLs (Tracking Protection Lists) [37] that aim to provide users with tools to restrict tracking and/or behavioral advertising could also mitigate the identified privacy threat. However, these solutions often prevent advertising companies from targeting ads or even serving ads to users.

There exist several possible countermeasures that could be used to target ads to users and mitigate the information leakage identified in this chapter. In particular, there are ongoing efforts to design and deploy privacy-preserving ad systems (e.g. Privad [39] and Adnostic [40]) whose main principle is to select ads locally. These solutions make the eavesdropping and filtering of targeted ads, and therefore our inferring attack, much more difficult. Another possible solution would be to send all ad requests and responses (containing DoubleClick cookies and ads content) over secure channels (SSL). However, we believe that this solution needs deeper analysis from the research community and the advertising industry since it might be too costly, not practical or altogether hard to deploy.

Stealing ads preferences via an active attack. The attack presented in this chapter is *passive*, i.e. completely transparent to the victim and to the ads providers. We note that a

user's preferences can also be stolen by a simple active attack. If fact, if an adversary is able to steal the victim's DoubleClick cookie, it can connect to his Google Ads preference page and retrieve his preferences. We examined the top 100 commercial websites from Alexa and found that at least 71% of them exchange DoubleClick cookie in clear with remote servers. Stealing a Double Click cookie is then quite easy. We implemented and tested this cookie hijacking attack, and were always able to retrieve the victim's Ads preferences page with a simple request to Google Ads servers. This attack is simple, however as opposed to our scheme, it is active and intrusive.

Retargeting ads. This chapter did not consider "retargeting ads"[6], which advertise the services or products of a site that a user has visited. Consider a user who is looking for a hotel in Vigo, Spain and performs some searches on the site `www.hotels.com`. It is very likely that he will consequently receive frequent ads advertising hotels in Vigo while browsing the Internet. Retargeting ads are not only targeting a particular user's interests, but specifically aim to match an exact intention or previous online action. Retargeting ads actually leak much more information about the user. In fact, in our example, they will not only leak that the user is interested in traveling, but also his actual destination i.e. Vigo, Spain. Note that retargeting ads are served independently of Google Ads Preferences profiles. A user will receive retargeting ads even if he empties his ads preferences profile. The only way to stop receiving retargeting ads is to clear his cookies or to completely opt out of targeted ads.

4.6 Summary

In this chapter, we showed that targeted ads contain valuable information that allows accurate reconstruction of users' interest profiles. We presented a methodology to categorize and filter targeted ads, which are in turn used to infer users' profiles. Based on both real users' web histories and synthetic users' profiles, we showed that our technique achieves a high accuracy in predicting general topics of users' interests. Additionally, using only a limited number of collected targeted ads we demonstrated that an adversary can capture on average more than half of targeted profiles. The algorithms described in this chapter are simple and probably not optimal. We believe they could be improved in many ways.

Many people claim that the main issue in online behavioral advertising is not related to ads personalization itself, which allows users to receive useful ads, but rather to the fact that it requires users' activities tracking. In this chapter, we show that ads personalization can also be harmful to users' privacy and does actually leak sensitive information such as users' profiles. We also note that this information leakage is not specific to online behavioral advertising, but in fact exists in any personalized content (news, searches,

[6]More details about retargeting ads are discussed in Chapter 6

recommendations, etc.). As the web is moving toward services personalization almost everywhere, special attention should be paid to these privacy threats.

Chapter 5

Privacy Leaks in Ad Exchange

Real-Time Bidding (RTB) and Cookie Matching (CM), the two common used protocols in ad exchange, are transforming the advertising landscape to an extremely dynamic market and make targeted advertising considerably permissive. The emergence of these technologies allows companies to exchange user data as a product and therefore raises important concerns from privacy perspectives. In this chapter, we perform a privacy analysis of CM and RTB and quantify the leakage of users' browsing histories due to these mechanisms. We study this problem on a corpus of users' Web histories, and show that using these technologies, certain companies can significantly improve their tracking and profiling capabilities. We detect 41 companies serving ads via RTB and over 125 using Cookie Matching. We show that 91% of users in our dataset were affected by CM and in certain cases, 27% of users' histories could be leaked to 3rd-party companies through RTB.

We expose a design characteristic of RTB systems to observe the prices which advertisers pay for serving ads to Web users. We leverage this feature to study how user profiles are valued by advertisers. Through our experiments, we confirm that users with known history are evaluated higher than new comers, that some user profiles are more valuable than others, and that users' intents, such as looking for a commercial product, are sold at higher prices than users' browsing histories. In addition, we show that there is a huge gap between users' perception of the value of their personal information and its actual value on the market. A recent study by Carrascal et al. showed that, on average, users evaluate the price of the disclosure of their presence on a Web site to EUR 7. We show that user's browsing history elements are routinely being sold off for less than $0.0005.

5.1 Introduction

Real Time Bidding (RTB) [126] is a novel paradigm of serving ads with the aim of bringing more liquidity to the online advertising market. When a user visits a Web site which

displays advertisements (ads) through RTB, the ad request is sent to an Ad Exchange which subsequently broadcasts it along with user data to *ad buyers* and holds an auction. These buyers bid in this auction and the winning party is allowed to serve ads to the user. The underlying technology to exchange users' identification data between Ad Exchanges and buyers is *Cookie Matching*, which allows two different domains to match their cookies of the same user.

Although RTB and Cookie Matching are acclaimed by the advertising industry, their privacy implications are not adequately understood. Cookie matching enables the possibility of linking the profiles of a single user in databases of two independent companies and is an integral part of RTB. In RTB, Ad Exchanges leverage Cookie Matching to broadcast user data to ad buyers. In other words, users' data become a product that is auctioned in real time in the online advertising market.

RTB-based spending is growing rapidly and is expected to account for more than 25% of the total display advertising sales in the US by 2015, up from 10% in 2011. By 2015, the majority of indirect display ad sales revenue will be traded using RTB in the United States and the most developed European markets [127]. RTB and Cookie Matching become increasingly rampant in the online advertising industry, yet to the best of our knowledge, there have been little academic studies of their privacy implications. In this chapter, we conduct an empirical study of these technologies and analyze how they impact users' privacy. We believe that it is important for users, researchers and privacy advocates to understand this privacy implication in very details.

While estimating value of user's private information is an interesting problem [128, 105], evaluating it is subtle and not obvious. Several recent research studies established results from the users' perspective [101]. Users, however, often do not have a developed sense of privacy. We approach this problem from the advertisers' perspective based on a market principle: *users' private data are worth as much as someone is willing to pay for them*. By leveraging a design feature of RTB systems, we are able to observe prices that advertisers pay for an ad impression after winning an auction. We utilize these prices to conduct a detailed analysis of the value of users' private data, with a focus on users' Web browsing history.

In summary, the main findings in this chapter include:

- We quantify the impact of Cookie Matching (CM) and Real-Time Bidding (RTB) on users' privacy. We show that CM happens very frequently and is performed by a large number of companies; some of them execute Cookie Matching in a significant proportion of the studied users' profiles (up to 91% of the 100 profiles we studied in our experiments). Our analysis of RTB shows that Ad Exchanges (e.g. DoubleClick) broadcast user-visited sites to a considerable number of bidders in real time; some of the bidders can learn up to 27% of users' histories through this mechanism.

- We provide an analysis of the value of users' private data from the advertisers' perspective based on prices they paid for serving ads to users. We confirm the fact that when a user's Web history is previously known to advertisers, they are willing to pay a higher price than in the case of new users. We also show that users' intents, such as browsing a commercial product, are higher valuated than their general histories, i.e. browsing sites not related to specific products. Finally, we highlight a huge gap between users' perception of the value of their personal information and its actual value on the market.

5.2 Background information

5.2.1 Cookie Matching

Cookie Matching (CM), an integral part of Real-Time Bidding, is a mechanism allowing two separate parties to synchronize their users' cookies [129]. For example, an Ad Exchange and a Bidder (ad buyer) normally attribute their own distinct cookies to the same user. After an execution of Cookie Matching protocol, one or both of them will have these cookies mapped to each other. Some Ad Exchanges, notably DoubleClick, create and use a unique user id (e.g. one-way hash of the cookie) instead of a cookie, with the aim to protect the actual cookie content from being revealed to the Cookie Matching partners. Nevertheless, we detected that many others are sending clear-text cookies for matching.

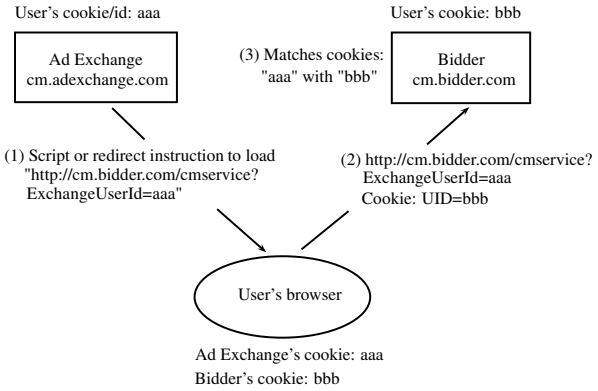


Figure 5.1: Cookie matching protocol

Figure 5.1 shows the main phase of Cookie Matching. Ad Exchange typically sends a script or a redirect instruction in order to instruct the user's browser to load a URL provided by the Bidder with the Ad Exchange's user's cookie/id in the parameter. The Bidder obtains the Ad Exchange's cookie/id upon receiving this request and matches this cookie/id with its own cookie. In some cases, this process can happen in the reverse direction, which results in the Cookie Matching on the Ad Exchange's side. Cookie matching is also known under different names, such as cookie syncing, pixel matching, etc. In this chapter, we use

"Cookie Matching" for all such actions of cookie synchronization between two separate entities.

5.2.2 Real-Time Bidding

Real-Time Bidding (RTB) [126] allows advertisers to buy online advertisement spaces at real-time through Ad Exchanges. Here we discuss the mechanism of DoubleClick's Ad Exchange [130], which is likely the most representative. Other Ad Exchanges, for example Pulse Point [131], employ similar approaches. The OpenRTB initiative [132], which aims at standardizing RTB, provides a similar description.

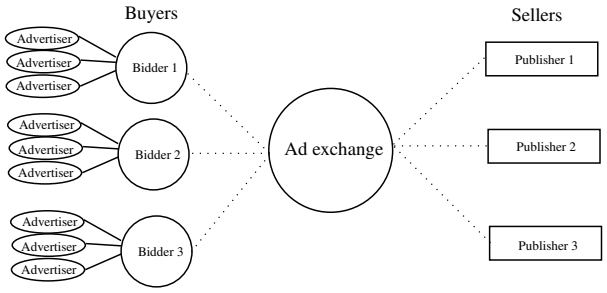


Figure 5.2: Ad Exchange model

The four main entities taking part in Real-Time Bidding include: 1) *Publishers* (e.g. `nytimes.com`), possessing the Web sites which display ads, 2) *Ad Exchanges* (e.g. DoubleClick), which enable ad transactions between ad sellers (the *publishers*) and ad buyers (the *bidders*) based on auctions held in real time, 3) *Bidders* (e.g. Criteo[1]), which are big advertising agencies representing small and medium advertisers to bid in RTB auctions in order to win ad spaces, and finally 4) *Advertisers* (e.g. hotels.com), which want to advertise and sell their products or services. Each time an ad is displayed in a Web site visited by a user, we call this event an *ad impression*. The RTB mechanism works as follows: When a user visits a publisher's Web site belonging to an Ad Exchange's advertising network, a HTTP request is sent to the Ad Exchange. The Ad Exchange subsequently sends *bid requests* for this ad impression to its bidders. Bidders then analyze the impression and submit their *bid responses*, which include prices they are willing to pay and their ad snippets. The bids are submitted to an online auction, and the Ad Exchange serves the winner's ad snippet on the user's visited Web site. After a successful transaction, the Ad Exchange charges the winning bidder, and pays the publisher after subtracting a commission. The total process generally happens in less than $100ms$.

Bid requests sent from Ad Exchanges to bidders typically contain information such as the Ad Exchanges' user's cookie (or user's id) and the user's visiting context including the following information: the URL of the Web site being visited, the categories of the

[1] `http://www.criteo.com`

site, the first three bytes of the user's IP address[2], various information concerning the user's browser and others [134, 135]. Upon receiving a bid request, the bidder finds its user cookie through the Ad Exchange's cookie thanks to Cookie Matching, provided that this protocol has been executed previously. It then determines the bid price based on the user's profile it possesses and the user's context provided by the Ad Exchange. Bidders can also bid on new users about whom they do not possess any prior information. When a bidder wins the auction, it has the right not only to serve ads, but also to initiate a Cookie Matching with the Ad Exchange.

An online Ad Exchange works similarly to a stock exchange [136], only trading in audiences for online ads. This mechanism helps publishers to sell their ads at the most competitive price, while allowing bidders to flexibly adjust their buying strategy in real time.

5.2.3 The Economics of Real-Time Bidding

The payment model used in Real-Time Bidding is Cost-per-mille impression (CPM) [137], which means every transaction through Ad Exchange is on a pay-per-impression basis. However, some advertisers might prefer the Cost-per-click (CPC) model [138], as its performance is more effectively measurable than CPM. As a result, a hybrid model exists, in which real-time bidders (e.g. Criteo) buy ad impressions from Ad Exchanges and sell ad clicks to advertisers. In this model, the bidders are expected to bid high enough in Ad Exchange in order to win the auction, while ensuring an adequate click probability to gain a margin benefit. Click probability depends largely on how the ad content matches user profiles and/or visiting contexts.

In this work, we aim to analyze how bidders evaluate users' personal data on behalf of advertisers. We therefore focus on analyzing the strategy from the advertiser's perspective. Advertiser's purposes normally include: 1) inviting users to their Web sites for buying a product or using a service, and/or 2) improving brand awareness. In both cases, the common goal is to reach potential customers. As most of Ad Exchanges encourage truthful bidding, for example by the use of Vickrey auctions [139], the best strategy for advertisers is expected to be bidding in accordance with the true value they can expect to get from the user.

5.3 Cookie Matching and RTB Detection

In this section, we describe the discovery techniques that we employed. First, we introduce the request hierarchy detection technique, which serves as a basis for the others. Second,

[2]Note that some companies, such as Pulse Point, actually send full IP addresses [133].

we present our technique to detect Cookie Matching. Then we describe the Real-Time Bidding detection technique, which is based on the discovery of winning prices.

5.3.1 Request Hierarchy Detection

We describe our technique to detect all causal relations between HTTP requests. The requests are often originating from Web sites' HTML tags including `<script>`, `<img>` or `<iframe>`. The responses to these requests might also contain HTML elements or JavaScript code that subsequently initialize other requests, and so on. Detecting such causal relations between requests is important to observe Cookie Matching and Real-Time Bidding events.

Assuming two HTTP requests A and B, A happening before B, our approach is as follows: we observe the *HTTP Referer* field in the request header of B (B's Referer), and *Location* field in the response header of A (A's Location). If A's Location contains B's URL, this means the browser redirects the request from A to B. Meanwhile, B's Referer containing A's URL means B is loaded from the content of A. Nevertheless, in the case of requests being dynamically initiated as a result of JavaScript scripts, the Referer field might not be a good indicator, as it points to the visited Web site rather than the source of the script. Therefore, we also scan all the JavaScript files we encounter during the loading of the site. If a request's URL is detected in a JavaScript script, we conclude that the script creates this request. However, this approach fails when JavaScript code builds URLs dynamically by concatenating dynamic parameters into a domain. We therefore also search JavaScript scripts for domains.

5.3.2 Cookie Matching Detection

In Cookie Matching, one domain synchronizes its cookie with another domain by including it in the request sent to the latter. For example, domain A returns a script to the browser which will invoke a request to domain B such as: http://B_URL?ExternalUserId=[A's cookie] (see section 5.2.1). Therefore, in order to detect Cookie Matching, we detect all the causal relationship $A \rightarrow B$, then scan all cookies from A and all parameters sent to B. We only take into account values that are sufficiently long, i.e. whose length exceeds 10 characters, as shorter strings are usually temporary values, unrelated to our research. If a match is detected, we consider it to be Cookie Matching. We manually checked a considerable number of values to confirm that they are indeed long-term cookies.

This method fails with DoubleClick, as this company uses a unique *user id* instead of the cookie itself. In this case, we leverage the Google's Cookie Matching protocol description [129], which clearly defines specific URL patterns. Examples of these URLs are presented in Table 5.1. In these URLs, `google_nid` is the unique id that Google

Table 5.1: Google's Cookie Matching URLs

Google's Cookie Matching URLs
http://cm.g.doubleclick.net/pixel?google_nid=[...]&google_cm
http://cm.g.doubleclick.net/pixel?google_nid=[...]&google_push=...

assigns to its Cookie Matching partner (ad buyer), while `google_gid` is the Google user id corresponding to the Google's user's cookie. Google distributes buyer-specific user ids, which means different buyers see different Google user ids for the same Web user.

5.3.3 Real-Time Bidding Detection

Bidders are charged for every ad impression won through RTB. The paid prices are usually included in the requests related to ad creatives which are served via Ad Exchanges with the help of a `WINNING_PRICE` macro. We detect RTB by interpreting the values of parameters in HTTP requests and looking for such price pattern.

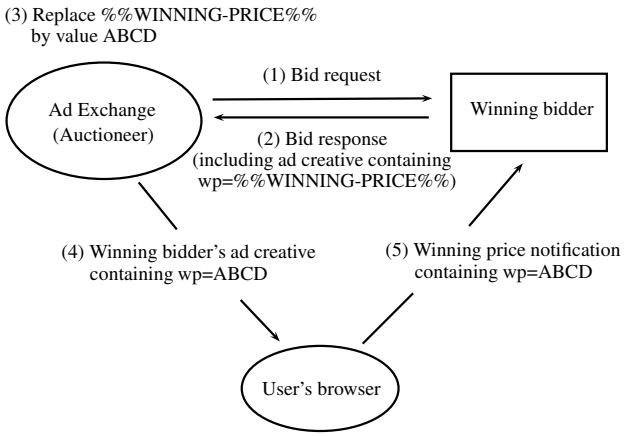


Figure 5.3: Winning price notification in Real-Time Bidding

The purpose of the `WINNING_PRICE` macro is to allow the Ad Exchange to notify the winning bidder about the actual price it has to pay for the ad impression[3]. The mechanism of winning price notification is shown on Figure 5.3. In its bid response sent to the Ad Exchange, each bidder includes its ad creative, i.e. a small HTML or JavaScript code responsible for displaying ads. The ad creative normally contains the winning price macro in a special text form (e.g. *%%WINNING_PRICE%%* in the case of DoubleClick) appended to a URL (which we call *ad_URL*). After the auction, the Ad Exchange replaces the winning price macro in the winner's ad creative with the actual winning price, and serves the creative to the user. Upon reception of this message, the user's browser runs

[3]This price is not necessarily equal to the actual bid price as most of Ad Exchanges use second-price auctions, in which the winner pays the *second highest* bid price incremented by a small pre-defined value.

the creative which initializes a HTTP request to the *ad_URL* in order to fetch the actual advertisement. Note that this HTTP request also contains in its parameters the winning price.

In the following, we describe the DoubleClick's *winning price format*. During the experiments we conducted in this work (section 5.4 and 5.5), we detected a considerable number of other companies apparently using the same or similar formats. In DoubleClick Ad Exchange, which belongs to Google, the price is encrypted and subsequently has a fixed length of 28 bytes (Figure 5.4). It is then encoded in a 38-character-length Web-safe Base64 [140].

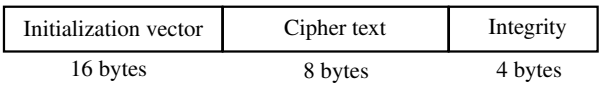


Figure 5.4: Google's winning price format

Each bidder shares a different *encryption key* and *integrity key* with the Ad Exchange to allow the decryption and verification of the encrypted price. The *initialization vector* contains a timestamp in the first 8 bytes with the aim to detect any stale response attack [140]. We rely on this timestamp to detect encrypted prices. Specifically, we extract all the suspected values of the URL's parameters, which have a length of 38 characters and contain only valid Web-safe Base64 characters, in each HTTP request initiated by the browser. We decode each of these values and extract the first 8 bytes to investigate whether this is a valid timestamp. According to Google's description, we convert the first 4 bytes to seconds and the last 4 bytes to milliseconds, and then obtain the total milliseconds. We compare this timestamp to the timestamp obtained from the response header of the investigated request. If they do not differ beyond a threshold, we consider the timestamp as valid, and assume the encrypted text is a valid price. We use a 5-minute threshold.

Each price is included in a URL as a value of a specific parameter. For example, the creative could include a URL in the following form: `http://bidder_URL?wp=[Winningprice]`, here `wp` being a URL's parameter whose value is the winning price. We used the winning price detection technique described previously to detect such forms of URLs. Table 5.2 provides some examples of the domains and the corresponding parameter names we encountered during our experiments and tests. For example, the price URL for Invite Media has the following form: `http://invitemedia.com?cost=[Winningprice]` (extra parameters stripped for clarity) – the price parameter is `cost` in this case.

When investigating requested URLs during our experiments in search for such patterns, we surprisingly found a substantial number of winning prices that were not encrypted. We deduce that these values are winning prices because of the following reasons. First, these URLs share identical patterns with URLs containing encrypted prices (same domain

Table 5.2: Clear-text price URL patterns

Domain	Parameter name
invitemedia.com	cost
mathtag.com	price
gwallet.com	win_price
adnxs.com	pp
mythings.com	rtbwinprice

name, same list of parameters), but include a clear-text value instead of an encrypted one for the same URL's parameter. Second, the values we obtained were very often in form of floating-point number (e.g. 0.5) or integer in micros format (i.e. 1 is converted to 1,000,000 micros), which match exactly the price format description of the advertising industry. Moreover, the parameters' names for these values are often contextual and meaningful. Examples include: "*win_price*", "*cost*", "*price*" or even "*rtbwinprice*" as shown in Table 5.2. In total, we detected 41 domain names (e.g. *ad.turn.com*) belonging to advertisers (*Turn* in this case), and corresponding HTTP parameters (*acp* in this case) whose values contained prices.

It is understandable that companies use the same URL patterns for winning price notification, regardless of the formatting of prices, while working with different Ad Exchanges. This helps them maintain a unified and simpler information system. The fact that a significant proportion of prices are in clear-text gives us an opportunity to observe how advertisers evaluate the value of each impression (see Section 5.5).

In summary, we use both encrypted and clear-text prices to detect Real-Time Bidding. It should be noted that winning price notification is not obligatory. Rather, it is an option for bidders and depends on the policy of Ad Exchanges. This means there might be some communications related to Real-Time Bidding that we could not detect. This happens if Ad Exchanges choose other schemes to notify the winning prices (e.g. server to server) or real-time bidders do not use the `WINNING_PRICE` macro. Therefore, the number of Real-Time Bidding communication we detected using this scheme can be considered as a *lower bound* of the actual number.

5.4 Cookie Matching and RTB Analysis

5.4.1 The RTBAnalyser Plugin

We implemented all the aforementioned techniques in a Firefox plugin, *RTBAnalyzer*, which is a modified version of HttpFox [141], an open source Firefox plugin. We implemented a Firefox `nsIObserver` interface to observe all HTTP requests and responses,

then applied the previously-described techniques to detect Cookie Matching and Real-Time Bidding. The plugin builds a hierarchy organization of all HTTP requests originating from the sites visited by the user. For each request, it collects the domain name (not the full URL) and identify whether the request is related to Cookie Matching or Real-Time Bidding. In case of Real-Time Bidding, it also collects the related winning prices. These analyzed information are saved into JSON format and sent to our server. It is important to note that each domain name of first-party sites contained in these data reports is replaced with a random value in order to protect privacy of plug-in users.

5.4.2 Dataset

We distributed the plug-in to our colleagues and friends and asked them to install it and browse the Web normally for a number of days. The experiment was performed during the month of June, 2013. The volunteers were mostly researchers and students based in our country of residence, France. Data were automatically sent to our servers every hour or at user request, depending on the chosen installation option. We did not attempt to create any link between the data we obtained and the personal identities of the users. As a result, we do not know who actually participated in the experiment. At the end of the experiment, we selected the top 100 profiles, after removing those that contained less than 70 sites. This dataset is used in sections 5.4.3 and 5.4.4, and part of section 5.5.

5.4.3 Cookie Matching Privacy Analysis

5.4.3.1 Privacy analysis

Companies normally build independent user profiles identified by their own cookies. Cookie Matching facilitates potential cooperation between these systems to exchange their users' data and possibly build larger user profiles. Without matching cookies, it would be difficult to link two profiles of the same user maintained by two separate entities. This results from the fact that trackers are usually able to see only the URLs a user is visiting and no other identifying information, such as e-mail address or user's name. While tracking and data exchange for advertising purposes are increasingly prevalent, technologies like Cookie Matching could potentially enable user tracking to a much larger scale.

5.4.3.2 Methodology

In order to demonstrate and quantify the potential risks described in the previous section, we studied the 100 profiles and identified the most active companies performing CM. Simultaneously we monitored the top trackers of these profiles, and evaluated the extent of potential history discovery by these entities via tracking. Finally, we evaluated to what

extent these companies could broaden their tracked users' profiles by making use of CM and sharing their knowledge of profiles.

5.4.3.3 Results

We first counted the cumulated numbers of Cookie Matching events following each site of the profiles in the real user dataset, and then averaged out these values. We show the results for the first 70 sites in the users' histories. Figure 5.5 displays these average values according to the number of visited sites. It shows that more than 60 Cookie Matching events happen when a user visits 40 sites (red curve) and more than 30 domains are involved (green curve). We can observe that the number of Cookie Matching increases regularly with the number of visited sites. The average cumulated number of cookie matching events after 70 visited sites is more than 100, performed by nearly 60 different domains on average. These results show that Internet users are encountering Cookie Matching at regular intervals.

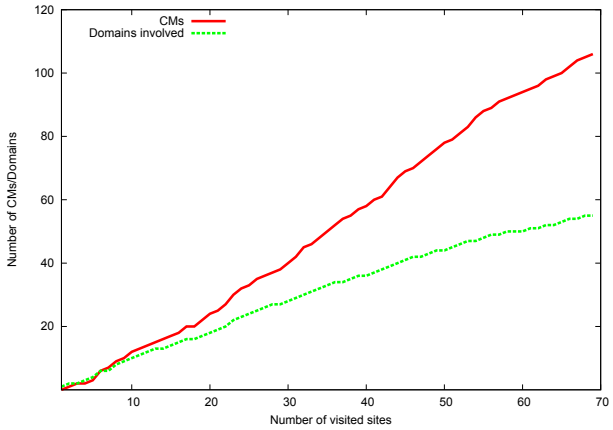


Figure 5.5: Cookie matching frequency

We observed the frequency of Cookie Matching performed by each pair of companies, and detected that many of them executed this scheme routinely. Table 5.3 shows the 20 pairs of domains that performed Cookie Matching the most. We noticed that Facebook (facebook.com) and AppNexus (adnxs.com) matched their cookies in 91% of profiles. The numbers are respectively 87%, 86% and 85% for the following pairs: Turn (turn.com) - Admeld (admeld.com), DoubleClick (doubleclick.net) - Rfihub (rfihub.com) and DoubleClick (doubleclick.net) - AppNexus (adnxs.com).

We investigated the *top* 25 *trackers*, i.e. the domains that tracked the largest parts of the studied users' histories. We detected these domains by capturing all outgoing requests when a user visited a Web site. If there was at least one request from this site to a 3rd-party

Table 5.3: Top pairs of domains executing cookie matching the most

Pair of domains	**Frequency (% profiles)**
facebook.com - adnxs.com	91
turn.com - admeld.com	87
doubleclick.net - rfihub.com	86
doubleclick.net - adnxs.com	85
doubleclick.net - mathtag.com	85
adnxs.com - admeld.com	84
doubleclick.net - turn.com	80
atdmt.com - bing.com	80
demdex.net - acxiom-online.com	79
doubleclick.net - yieldmanager.com	77
invitemedia.com - admeld.com	73
mathtag.com - admeld.com	71
doubleclick.net - invitemedia.com	71
doubleclick.net - amazon-adsystem.com	70
rubiconproject.com - rfihub.com	70
adnxs.com - amazon-adsystem.com	69
adnxs.com - rfihub.com	68
turn.com - p-td.com	67
turn.com - rubiconproject.com	65
mathtag.com - facebook.com	64

domain with Referer field in the HTTP header containing the site's URL, this domain is considered *being aware* of this visit. Table 5.4 shows the top 25 trackers with their average percentage of tracked user's history that we detected in our real user dataset. We observed that among the companies in the top 20 pairs of companies using Cookie Matching most frequently (Table 5.3), 56% of them are in our list of 25 top trackers (Table 5.4). Meanwhile, 36% of these top trackers are in the top 20 pairs of companies most often performing Cookie Matching. These results show that, although Cookie Matching is used by numerous companies, the top trackers are often more involved than others.

We detected that some companies in the list of 25 top trackers can considerably increase the size of their users' profiles if cooperating. For example in our experiments, Facebook and AppNexus respectively tracked 31.55% and 17.4% of a user's history on average, and they performed CM in 91% of the studied profiles. Their total Web history coverage would increase to 39.35%, on average, if they were merging their user histories. Table 5.5 shows some examples of the potential combined profile sizes in cases of other companies. In this

Table 5.4: Top trackers

Tracker	**Average (% of user history)**
google-analytics.com	56.38
doubleclick.net	50.72
scorecardresearch.com	38.57
facebook.com	31.55
google.com	24.92
googleapis.com	23.84
facebook.net	23.44
quantserve.com	23.17
twitter.com	22.65
googleadservices.com	20.47
googlesyndication.com	20.41
2mdn.net	18.17
fbcdn.net	17.76
gstatic.com	17.56
adnxs.com	17.4
imrworldwide.com	15.73
yieldmanager.com	13.39
cloudfront.net	11.11
bluekai.com	10.92
atdmt.com	10.24
invitemedia.com	10.09
googletagservices.com	9.39
turn.com	9.21
rubiconproject.com	8.65
mathtag.com	8.01

table, Q_1, Q_2, Q_3 are the first, second, and third quantiles respectively, computed among the 100 studied profiles.

A case study of Google and DoubleClick. Based on the results from Table 5.4, Google possesses 8 domains belonging to the top trackers: google-analytics.com (56.38%)[4], doubleclick.net (50.72%), google.com (24.92%), googleapis.com (23.84%), googleadservices.com (20.47%), googlesyndication.com (20.41%), gstatic.com (17.56%), googletagservices.com (9.39%). Although cookies used for these domains are all different, it is

[4]Even though Google Analytics uses unique cookies per sites, it is potentially possible to link these cookies across sites, for example by leveraging user's IP address

Table 5.5: Potential percentage of profile tracked after combination. Averages and *i*th quantiles.

Domains	**Avg. (%)**	**Q_1 (%)**	**Q_2 (%)**	**Q_3 (%)**
doubleclick.net - adnxs.com	52.43	48.74	52.86	56.34
doubleclick.net - yieldmanager.com	52.01	48.54	52.7	56.82
facebook.com - adnxs.com	39.35	36.0	39.47	44.3
adnxs.com - amazon-adsystem.com	19.32	16.05	18.67	22.35
invitemedia.com - admeld.com	14.12	11.84	14.29	16.44

trivial to match them, for example by inspecting the IP address. By combining all data tracked by these domains Google could possibly know 80.13% of a user's visited sites, on average.

DoubleClick's Cookie Matching services are utilized by a substantial number of 3rd-parties. By analyzing Cookie Matching communications in all our experiments and tests[5], we extracted the host names of DoubleClick's Cookie Matching partners and counted their distinct top-level domain names. For example, *dis.ny.us.criteo.com*, *dis.jp.as.criteo.com* and *dis.eu.criteo.com* share the same top-level domain name, *criteo.com*, and were counted once. In total, we detected 125 top-level domains performing CM with DoubleClick. It is interesting to note that one of the detected domain names was `e.visualdna.com` which belongs to a Big Data analytics company specializing in psychometrics, Visual DNA[6]. This example shows that Cookie Matching is not only used by advertisers, but also by other entities.

5.4.4 Real-Time Bidding Privacy Analysis

5.4.4.1 Privacy analysis

By combining RTB and CM, users' private data could potentially be leaked to bidders involved in real-time auctions. Figure 5.6 illustrates this leakage. We assume a situation between an Ad Exchange *ADX* holding the real-time auction, and a set of bidders $B_1, ..., B_k$ registering for the auction. Whenever a user visits a Web site W which requests ads from *ADX*, *ADX* sends a bid request to all the bidders. The bid request includes W's URL, the ADX's cookie of the user along with other additional information as discussed in section 5.2.2. In our study we focus on the leaks of browsing histories, although it is evident that the additional information can potentially be used to fingerprint the user's browser [142]. Each time a bidder receives a bid request, he can save the W's URL and the *ADX*'s cookie, resulting in a list of URLs assigned to each specific cookie from the *ADX*, provided that several bid requests containing this cookie were seen previously. Whether Cookie

[5]Including all experiments we conducted in this work (section 5.4 and 5.5)

[6]`http://www.visualdna.com`

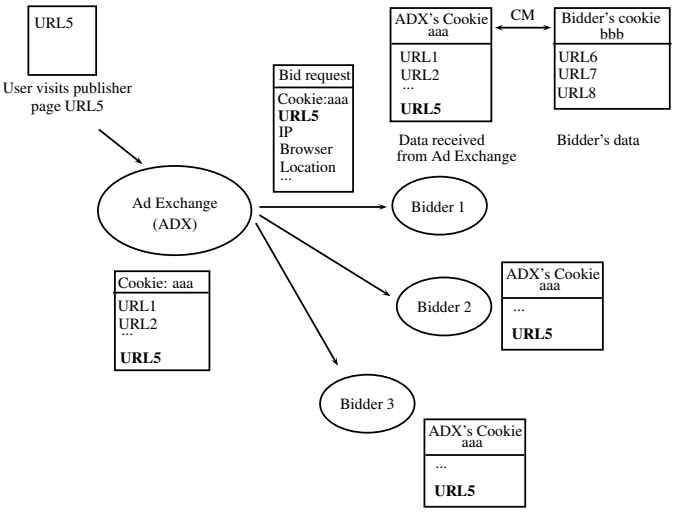


Figure 5.6: Information leakage in Real-Time Bidding

Matching takes place before or after these RTB processes, the bidder can combine all the previously-observed users' visited sites from received bid requests with its own user's profile identified by its own cookie. Even if Cookie Matching does not happen, these URLs can still provide a significant amount of information about the user identified by the *ADX*'s cookie.

5.4.4.2 Methodology

We aim to show the frequency of RTB communications and quantify the information leakage described in the previous section. We examined all Real-Time Bidding requests, which we detected using our RTB detection technique (section 5.4.1), in our 100 profiles and extracted the related Ad Exchanges and winning bidders. The winning bidders were identified by the domain of each request, while the Ad Exchanges by the domain of the parent request in the request hierarchy (as discussed in 5.3.1). We obtained a list of Ad Exchanges and for each Ad Exchange, a list of its bidders that won at least one auction. Examples of winning bidders in the case of DoubleClick Ad Exchange include AppNexus, AdRoll and InviteMedia.

We examined all profiles in the real user dataset. If a RTB event was detected on a site of a given profile, we assumed that all bidders participating in the RTB auction received the site's URL via the bid request sent by the Ad Exchange. We obtained the list of bidders associated to a given Ad Exchange by the use of methods described in the previous paragraph. We analyzed all sites in all the profiles and we counted how many sites would be leaking to each bidder via this mechanism. Subsequently, we divided the numbers of leaking sites by the total number of sites in the profile in order to quantify the history leakage.

It should be noted that each bidder can bid on several RTB Ad Exchanges, hence possi-

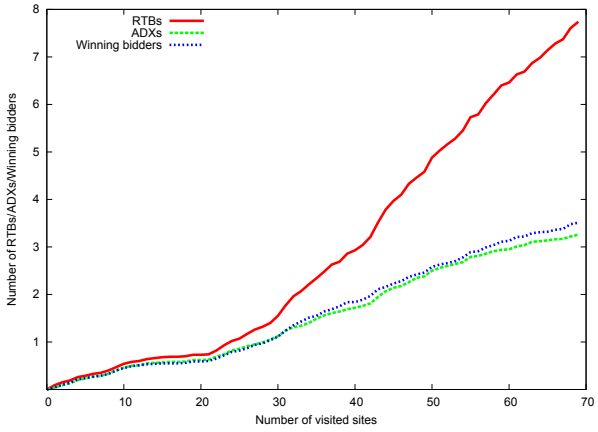


Figure 5.7: Real-Time Bidding frequency

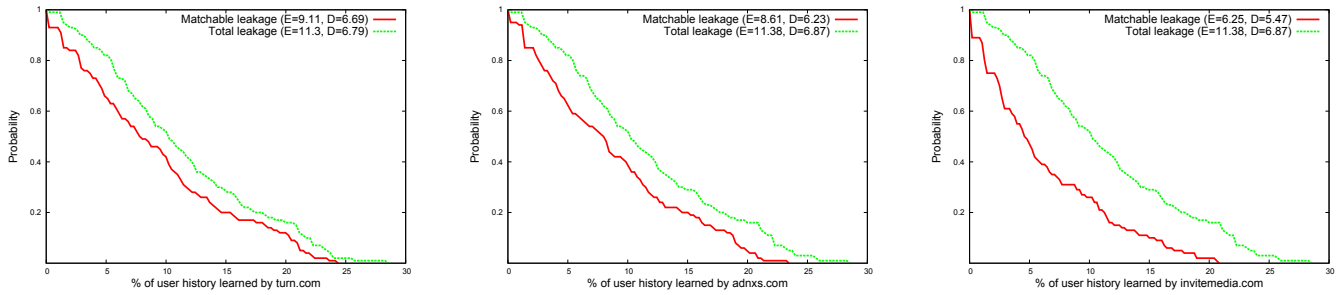
bly learn parts of a user's browsing history from each of them. For example, we detected that AppNexus bids simultaneously on DoubleClick's and Admeld's RTB auctions.

We considered all the URLs that an Ad Exchange possibly sent to a bidder (detected by the above mechanism) as the *total leakage*. However, if a Cookie Matching event was detected between the Ad Exchange and the bidder during the experiment, we considered the URL leakage as a *matchable leakage*, otherwise *unmatchable leakage*. In matchable leakage, bidders can obviously combine profiles obtained from Ad Exchanges with their own users' profiles using Cookie Matching (Figure 5.6). Meanwhile, in *unmatchable leakage*, it is not clear whether the Cookie Matching will happen in the future, or other techniques can be used to link the two profiles. We therefore consider that the leakage is less severe in this case. *Total leakage* comprises both these two cases.

5.4.4.3 Results

Figure 5.7 shows the average cumulated number of RTB events, distinct Ad Exchanges and winning bidders after each visited site in our profiles. The cumulated numbers of RTB events after n visited sites are averaged from those numbers computed for each profile (red line). The average cumulated number of distinct Ad Exchanges and distinct winning bidders are shown in green and blue respectively. The figure shows that, when considering web histories of size 70, RTB occurred in 10% of the sites.

Figure 5.8 presents the percentage of user's history the three companies, Turn, AppNexus and InviteMedia, could obtain from Ad Exchanges in RTB. The figure shows a Complementary Cumulative Distribution Function (CCDF) of the percentage of user's history leak among the 100 profiles as well as their average (E) and standard deviation (D). The blue line represents the CCDF for the total leakage, while the red one represents the



(a) Turn (b) AppNexus (c) Invite Media

Figure 5.8: CCDF of the percentage of user's history that bidders learned through RTB

matchable leakage. The total leakage on average is around 11% of a user's history, but can be as high as 27% for certain profiles. With such high percentage of history received through RTB, even without Cookie Matching, these companies can maintain a meaningful profile of a user. The matchable leakage is slightly lower, with the average value around 8% of the user's history. Bidders can easily combine these matchable data to their own users' profiles using Cookie Matching.

These numbers show that the user history leakage through RTB is significant. Given the fact that we only detected the lower bound of Real-Time Bidding communications, and that our assumption for the leakage is restricted to the bidders who won at least one observable auction, the leakage is potentially much higher in reality. Also, due to the rapid growth of RTB [127], these numbers are expected to considerably increase in the foreseeable future.

Information dissemination in RTB. We detected 41 winning bidders for all Ad Exchanges in total. In the case of DoubleClick Ad Exchange, we detected its 20 winning bidders, and 125 Cookie Matching partners which are likely real-time bidders as well. Although we did not encounter PulsePoint's Ad Exchange in all our experiments and tests[7], we found from its description a list of 59 RTB bidders [143]. These numbers suggest that 20-125 bidders might receive Web users' information in the case of DoubleClick, and at least 59 in the case of Pulse Point, which potentially constitutes a considerable information leakage.

5.5 Value of User Privacy

The observation of clear-text prices allows us to study how much advertisers pay for serving ads to users. As discussed in section 5.2.3, we believe that the prices paid in Real-Time Bidding reflect how bidders estimate the value of users. It is important to note

[7]Including all experiments we conducted in this work (section 5.4 and 5.5)

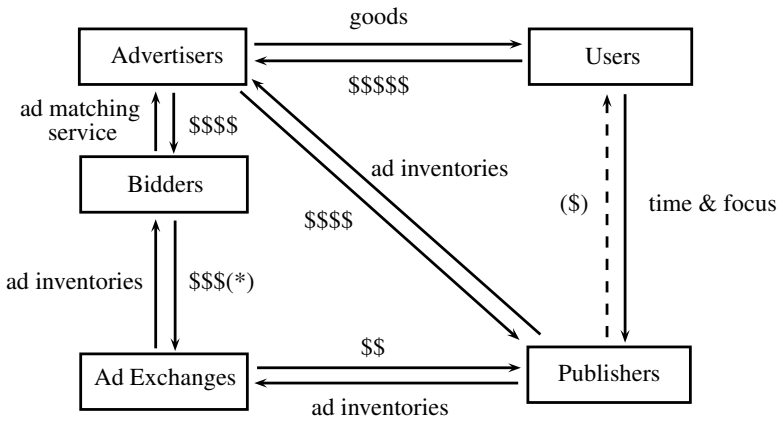


Figure 5.9: Monetary flows in advertising systems. The communication we monitored is indicated by (*). Source: [1].

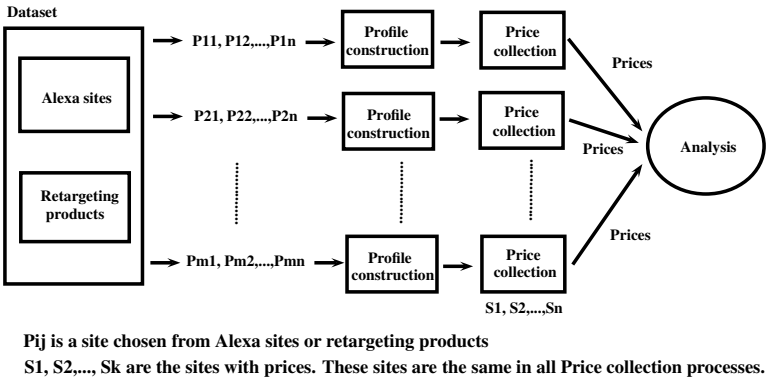


Figure 5.10: Experiment for artificial profile analysis

that all prices reported in this section are represented in CPM (Cost-per-mille impressions), which means each price is for 1,000 ad impressions. For example, a price of $0.12 CPM or $0.12 without any further explanation is actually $0.00012 per impression.

Figure 5.9 is a slightly modified version adopted from [1] and it shows the different monetary flows in a simplified model of advertising systems. The prices we retrieved for the analysis in this section are paid by *Bidders* to *Ad Exchanges* (marked with *(*)*).

In this section, we study whether the user's Web browsing history (or *profile*) affects prices that advertisers pay for serving ads[8]. Our methodology is to create a number of artificial profiles, use them to visit the same set of Web sites, collect and analyze winning prices. The profiles are created considering two aspects: *history categories* (i.e. categories of visited sites)[9] and *intents* (e.g. browsing for a commercial product). Figure 5.10 summarizes our experiment.

[8]A more complete study appears in our paper [144] which also investigates whether the visiting context (e.g. currently visited sites) affects the prices. In scope of this thesis, we only consider privacy-related factors.

[9]History categories can be used, for example in Google's and Yahoo's systems, to personalize ads [52][145].

5.5.1 Considerations

Encrypted prices. Our analysis is based on clear-text prices we were able to detect[10]. Despite the fact that we do not take into account encrypted prices, the prices we retrieved are comparable to the ones obtained directly from some Ad Exchanges' internal data and reported in other work [136, 1]. This constitutes a good evidence that encrypted and plain text prices are similar. In addition, we do not see any reason for advertisers to pay different prices on the basis of the price notification being encrypted or not.

Currency. In our analysis, we assume that the currency used by different companies is USD. Our assumption is based on the fact that the majority of Ad Exchanges are US-based, which is also observable in our dataset, and that USD is the most commonly used currency in international business. Some Ad Exchanges, e.g. Pulse Point, publicly state the use of USD as the only currency in their RTB protocol description [131]. Since bidders/advertisers often reuse the URL patterns (the same domain and parameter names) to receive price notification from different Ad Exchanges, regardless of the price format, they are likely using the same currency. Finally, the value range of prices detected in our experiments is similar to those presented in other work leveraging internal advertisers' data [136, 146], which mention prices solely in USD.

Tracking. Although there are many means of tracking the Web users, such as based on monitoring of IP addresses or fingerprinting techniques, cookie-based approach is still the dominant one. A good example is that Cookie Matching is a common technique used in RTB to match user profiles between two separate entities. Moreover during our experiments, we verified that targeted advertisements, for example ads about commercial products that users browsed previously, generally disappeared after clearing browser's cookies. In our work, we therefore assume that advertisers mostly rely on cookies to track users. Furthermore we assume that after clearing all the cookies of a browser, subsequent trackers perceive a request made by this browser as originating from a new user.

5.5.2 Methodology

In order to visit sites, we used Selenium [147] to instrument a Firefox browser equipped with RTBAnalyzer plug-in (described in section 5.4.1).

5.5.3 Dataset

We crawled 50 top Alexa sites from each of the following Alexa categories: *Adult, Arts, Business, Business-Financial Services, Computers, Games, Health, Home, Kids and Teens,*

[10] About 25.71% of the prices we collected were sent in clear-text.

News, Recreation, Science, Shopping, Sports. Those sites were used to construct history categories in each profile.

For the intent construction, we used three commercial Web sites that are very popular in our country of residence (France): fnac.com (electronic products), hotels.fr (hotel booking) and maty.com (jewelry). We call them *retargeting sites* hereafter, as after visiting them, the previously-browsed products from these sites often appear in online ads during regular browsing. We randomly chose 5 products from each of these sites and kept these three lists of products to build users' intents.

We extracted from the top Alexa sites a list of sites which often resulted with ads containing clear-text prices[11]. Among these sites, we extracted the top 17 sites which had the highest rate of clear-text price occurrence[12]. We call them *sites with prices*.

5.5.4 Experiment Description

We created 14 profiles for each of the following types:

- *New user*: empty profile
- *Only category*: only visit Alexa sites belonging to one category
- *Category + fnac.com*: visit Alexa sites belonging to one category, then visit 5 products on fnac.com
- *Category + hotels.fr*: visit Alexa sites belonging to one category, then visit 5 products on hotels.fr
- *Category + maty.com*: visit Alexa sites belonging to one category, then visit 5 products on maty.com
- *Only maty.com*: only visit 5 products on maty.com

We simultaneously ran 6 instances of Firefox browser to perform the tests with these profiles. Each instance was devoted to one kind of profile. For each profile in each browser's instance we subsequently performed a profile construction and price collection, and repeated this phase 10 times. All price collection processes were performed using the same set of *sites with prices*. We executed our experiments evenly throughout the day to ensure that time of day did not affect prices.

5.5.5 Results

We obtained 20 prices per profile and per round, consequently about 200 prices per profile in total (after 10 rounds), on average. The detected average prices per profiles are shown

[11] A sample of this list is available at `http://yourvalue.inrialpes.fr`.

[12] Examples of such sites are `accuweather.com`, `tinyurl.com` or `technorati.com`

Table 5.6: Artificial profile analysis. Prices in CPM.

Category	New user		Only category		Category + fnac.com		Category + hotels.fr		Category + maty.com		Only maty.com	
	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
Adult	N/A	N/A	0.44	0.20	0.56	0.27	0.64	0.20	1.12	0.21	N/A	N/A
Arts	N/A	N/A	0.51	0.17	0.52	0.15	0.66	0.23	1.28	0.29	N/A	N/A
Business	N/A	N/A	0.55	0.22	0.63	0.21	0.61	0.21	1.10	0.34	N/A	N/A
Business - Finan. Serv.	N/A	N/A	0.59	0.20	0.68	0.24	0.88	0.28	1.31	0.31	N/A	N/A
Computers	N/A	N/A	0.48	0.21	0.57	0.20	0.70	0.25	1.18	0.14	N/A	N/A
Games	N/A	N/A	0.80	0.35	0.74	0.29	0.81	0.40	1.41	0.27	N/A	N/A
Health	N/A	N/A	0.67	0.47	0.68	0.34	0.81	0.43	1.21	0.30	N/A	N/A
Home	N/A	N/A	0.58	0.21	0.70	0.39	0.57	0.23	1.00	0.23	N/A	N/A
Kids and Teens	N/A	N/A	0.64	0.33	0.65	0.27	0.74	0.29	1.25	0.27	N/A	N/A
News	N/A	N/A	0.50	0.12	0.72	0.38	0.74	0.29	1.09	0.18	N/A	N/A
Recreation	N/A	N/A	0.55	0.21	0.64	0.32	0.69	0.16	1.12	0.22	N/A	N/A
Science	N/A	N/A	0.50	0.19	0.60	0.37	0.59	0.21	1.36	0.24	N/A	N/A
Shopping	N/A	N/A	0.53	0.22	0.61	0.27	0.65	0.25	1.21	0.23	N/A	N/A
Sports	N/A	N/A	0.71	0.47	0.59	0.29	0.62	0.17	1.17	0.21	N/A	N/A
Average	0.41	0.10	0.58	0.26	0.64	0.29	0.69	0.26	1.20	0.25	1.17	0.26

in Table 5.6. The prices for profiles "*Only category*" are about 40% higher than those for "*New user*". Among "*Only category*" profiles, different profile categories result in different prices. This is particularly acute for the category *Games*, which exhibits prices 38% higher than average. Other category profiles with prices larger than the average price are *Sports*, *Health*, and *Kids and Teens*. Our results show that the type of visited sites is actually affecting prices that advertisers paid for serving ads to users.

The results indicate that retargeted ads (the ones which match users' intents) often receive higher prices than those for "*Only category*". These prices also differed among different retargeting advertisers. For example, prices from `fnac.com` were the lowest, with average $0.64, prices related to `hotels.fr` were slightly higher with $0.69, whereas `maty.com` had the remarkably highest average price of around $1.2. This could be explained by the strategies of the different advertisers and possibly by the prices of the advertised products. Interestingly, we also noticed that maty.com retargeted ads were displayed much more frequently than fnac.com or hotels.fr ads. Finally, we observed negligible differences in prices between "*Only maty.com*" and "*Category and maty.com*". This clearly shows that even though users' browsing histories are taken into account when advertising a product, advertisers actually value users' intentions much more.

Although we expected that retargeted ads are related to higher prices, the striking difference between winning prices for ads after visiting `maty.com`'s products and those after visiting non-retargeting sites deserves a detailed analysis. To our knowledge, most of ad auctions apply the *second-price principle* which means the winning bidder only pays a slightly higher price than the second highest bid price[13]. In other words, the paid price is

[13]Note that RTB systems can fine-tune their internal auction parameters to effectively switch from

the second highest price incremented by a small value defined by the RTB. Assuming that the average winning price of a "normal ad" is $0.4 and a retargeted ad has a significantly higher bid price of $1.2, the average winning price should still be close to $0.4. However, we observed much higher prices in the case of retargeted ads, specifically in the case of advertisements from `maty.com`, when served by Criteo. A possible explanation is that there are several competing retargeters who bid for this ad impression. In order to verify this, we conducted the following experiment: We used Ghostery [35] to block all other trackers except Criteo and a selected number of RTB systems (Admeld, AppNexus, Pubmatic and Rubicon; Criteo bids in their auctions). Similarly to the previous experiment, we browsed 5 products from `maty.com` and then a list of *sites with prices*. As described above, we blocked most of the trackers while browsing products on `maty.com`. We then stopped blocking trackers when visiting *sites with prices* when we aimed to detect the clear-text prices of advertisements. We performed this experiment 10 times. The average price observed during this experiment was $0.44 CPM, much lower than previously when the average was $1.17 (Table 5.6). In this setting Criteo could still win the auctions but at a much smaller cost (because we intentionally blocked the competitors). This result proved that other bidders had been involved in the initial scenario, and that retargeting companies are also competing on retargeted ads.

Advertisers also bid on new users. This could give them an opportunity to perform a Cookie Matching on them. As described in section 5.2.2, the winner has the right to initialize a Cookie Matching with the Ad Exchange when serving ads through Real-Time Bidding mechanism. The price $0.0004 per impression (average value $0.41 CPM divided by 1000) would be very reasonable for the opportunity to track a new user.

The relatively large variance values in the results can be explained by the fact that ads prices depend on several parameters such as different campaigns or different bidders. Furthermore, RTB is by definition dynamic, thus consequently auctions could possibly be won by different advertisers in each round of our tests. It is also important to note that the variance for *new users* is much (2.6 times) lower than in the other cases.

5.5.6 Real Profile Analysis

We analyzed clear-text prices obtained in the real user dataset (section 5.4.2). The results are shown in Table 5.7. Among the 100 users, 89 had at least one clear-text price. The average number of clear-text prices per profile is approximately 4. There is a high rate of variation among the prices per analyzed profile, with minimum at $0.04, maximum at $1.98, and average value of $0.43. We also investigated the 8 profiles which had at least 7 prices per profile to analyze how prices vary among them. Table 5.8 presents the number

second-price to first-price auctions [136]. However, as we show, this does not apply to our case.

Table 5.7: Real profile analysis. Prices in CPM.

Property	Value
Number of profiles with clear-text prices	89
Avg. number of prices per profile	3.83
Average price per profile	0.43
Standard deviation (price per profile)	0.37
Min price per profile	0.04
Max price per profile	1.98

Table 5.8: Real profile examples. Prices in CPM.

Index	Average price	Standard deviation	Count
1	0.16	0.17	7
2	0.26	0.14	11
3	0.41	0.51	8
4	0.43	0.20	8
5	0.45	0.31	8
6	0.91	0.68	13
7	1.11	0.89	7
8	1.13	1.00	8

(count), average value and standard deviation of prices in 8 profiles from our dataset. The prices we observed with real user profiles actually vary within the value range of prices detected in artificial profiles as shown in Table 5.6.

5.6 Discussion

5.6.1 Data Exchange between Companies

Data exchange is a growing trend in modern advertising systems. When targeted ads become increasingly sophisticated, the users' dataset maintained by an intermediary (e.g. an ad network) might not adequately meet these demands. Naturally, advertisers desire to target users with the use of their own data as well. For example, Facebook has been working with data vendors Datalogix, Epsilon, Acxiom and BlueKai [148] in order to allow its clients to serve ads based on their offline data [149]. RTB services enable advertising companies to use their own online data for serving targeted ads. While this data exchange is expected to enhance advertising performance, it should be designed with careful consideration. Otherwise, this could lead to users' data leakage between various

companies and the resulting loss of control over this data. In this chapter, we showed that this might indeed be the case with RTB. We investigated and quantified the leakage based on the assumption of non-adversary parties. With malicious attempts, e.g. collusion between the companies with the aim to combine their users' profiles, the risks could be much more severe.

5.6.2 Privacy-Preserving Targeted Advertising

There have been a considerable number of research work towards designing a targeted advertising system not utilizing tracking, such as Privad [39] and Adnostic [40]. Yet, most of the proposed solutions are designed in the traditional ad network setting. Their common idea is to save users' profiles on the client side; ad networks send coarse-grained ads to the client, which then can locally select the most appropriate ones to display, according to the user profile. It is not clear if these systems can be adapted to new technologies such as RTB. In RTB, the advertisers want to customize their bids towards each individual user, e.g. a jewelry advertisement could have different values when showing to a male and a female. Moreover, advertisers are likely interested in adjusting their buying strategy at real time. The emergence of such new demands and techniques requires a significant change in the proposed privacy-preserving targeted advertising systems, or even a new design approach, in order to address privacy problems while maintaining current business models.

5.6.3 The Economics of Private Data

In a study performed by Carrascal et al [101], users evaluate the disclosure price of their presence on a Web site to EUR 7, on average. In this work, we showed that this information is actually being sold off at a much lower price by Ad Exchanges and that its price depends on the user's browsing profile, in addition to other contextual information. Our experiments demonstrated that, on average, the presence of a user on a Web site is sold to the winner of the RTB auction for less than $0.0005 ($0.5 CPM). We also note that since the presence of the user on a Web site is actually broadcast to all the bidders during a RTB request, this cost can be shared among them. The actual cost per bidder could then be computed by dividing $0.0005 by the number of bidders, which we estimated to $20 - 125$ for DoubleClick. We acknowledge that the cost also includes the price paid for the ad delivery. The huge gap between these figures and those from the users' perception can be explained by the fact that user information is currently extremely easy to collect (e.g. by simply placing a small JavaScript code in a Web site), therefore could be sold at very cheap price.

Revenue per user. Estimating how much advertisers spend on a user is an interesting problem, and we aim to provide a rough estimation of this cost. According to the work of Castelluccia et al. [150], targeted ads account for about 30% of total ads. From the

analyses in the previous section, we can assume that the average price per ad is $0.0005.

We manually counted advertisements on 50 sites corresponding to an one-day browsing history of a volunteer and detected 40 ads in total (0.8 ad per site). We therefore derived the total number for targeted ads at around 12 (30% of ads on 50 sites) in the analyzed case. Setting the average price per ad to $0.0005, these ads cost advertisers $0.006 per day. Accordingly, the cost is approximately $0.18 per month and $2.16 per year. If, for example, Ad Exchanges take a commission fee of 20% for each transaction, they could earn around $0.432, and the publishers $1.728, per user, per year.

This simplified scenario is only meant as a rough estimation since many aspects remain uncertain. For example, the number of ads per site and the number of browsed sites per day may not be representative. We also assumed that all other cost models such as Cost Per Click (CPC) or Cost Per Action (CPA) can be converted to the equivalent CPM. For example a price of $0.01 CPC for an ad with click probability of 10% can be converted to a $1 CPM. We therefore assumed in our estimation that the average CPM price (established in previous sections) applies to all targeted ads. By this estimation, we showed an initial quantification of how much a user costs, or how much money which entities (Ad Exchanges, publishers, etc.) gain from the user's data in online advertising market.

5.7 Summary

In this chapter, we characterized Real-Time Bidding (RTB) and Cookie Matching (CM), and highlighted the core privacy risks associated with the use of these technologies. We showed that RTB and CM are observably prevalent on the Web and lead to significant user information leakage. Concretely, RTB can leak as much as 27% of a user's Web browsing history to a bidder involved in Ad Exchanges' auctions. The actual leakage is expected to be higher, since we only established a lower bound of actual RTB communications. The process is inherently non-transparent, and this *invisible* leakage cannot be observed using current tracking measurement tools such as Collusion [151] and Ghostery [35]. Nevertheless, a strict privacy protection approach, such as blocking all ad-related URLs using Ghostery or AdBlock Plus [34] could potentially solve this privacy problem.

RTB creates a data market where users' browsing data are sold at auctions to advertisers. We showed that advertisers are evaluating each individual user differently depending on several criteria. Our results indicate that the presence of a user in a Web site is often sold off for less than $0.0005, which is far lower than that from users' perception [101]. We highlight that such sophisticated methodologies being used to commoditize users' data without their awareness, let alone consent, is a problem that needs due attention.

Part II

Privacy-Enhancing Solutions

Chapter 6

A Practical Solution: Retargeting Without Tracking

Retargeting ads are increasingly prevalent on the Internet as their effectiveness has been shown to outperform conventional targeted ads. Retargeting ads are not only based on users' interests, but also on their intents, i.e. commercial products users have shown interest in. Existing retargeting systems heavily rely on tracking, as retargeting companies need to know not only the websites a user has visited but also the exact products on these sites. They are therefore very intrusive, and privacy threatening. Furthermore, these schemes are still sub-optimal since tracking is partial, and they often deliver ads that are obsolete (because, for example, the targeted user has already bought the advertised product).

In this chapter, we present the first privacy-preserving retargeting ads system. In the proposed scheme, the retargeting algorithm is distributed between the user and the advertiser such that no systematic tracking is necessary, more control and transparency is provided to users, but still a lot of targeting flexibility is provided to advertisers. We show that our scheme, that relies on homomorphic encryption, can be efficiently implemented and trivially solves many problems of existing schemes, such as frequency capping and ads freshness.

6.1 Introduction

6.1.1 Context and Motivation

Since behavioral targeting helps advertisers optimally allocate their advertising resources to their most likely potential customers and consequently increase their revenue, companies have been constantly improving their tracking and ad personalizing technologies with the aim to enhance targeting performance.

Retargeting ads have been introduced in recent years with the aim to match the exact user attention or previous online action. For example, a user who has visited hotels.com looking for a hotel in Paris will very likely receive frequent ads about this hotel during his subsequent browsing sessions, for instance on accuweather.com. Advertisers (hotels.com in this case) aim to bring these customers back to their sites by showing ads related to the products they previously showed interest in. Retargeting ads have been shown to be significantly effective; Criteo in particular confirmed that personalized retargeting ads perform 6 times better than general ads [152]. Increasingly, retargeting is becoming prevalent in travel, real estate and financial services industries [153].

Retargeting advertisers, mostly commercial online stores (e.g., hotels.com), often leverage a third party, called *retargeter* (e.g., Criteo), to handle the retargeting task. Retargeters track users on these stores to collect products that they are interested in, and then select one to advertise to a user when (s)he visits an ad-enabled website. This is beneficial to advertisers as they can outsource the optimization of the whole advertising process to an external party with dedicated resource and expertise. However, as users' interested products are centrally collected by third-party retargeters, this also poses significant privacy threats. For example, these products are shown to reveal users' important events in their life, such as being pregnant, getting divorced or graduating [58]. Compared to conventional tracking, where ad networks usually only collect urls of sites visited by a user (e.g., hotels.com) to infer his interest categories (e.g., traveling), retargeting trackers also retrieve *exact products* on each page thereby inferring more accurate information about the user (e.g., the city where he is searching for a hotel).

There have been serious public concerns about the prevalence and resulting privacy threats of retargeting. As observed by the author of a The New York Times's article [153]: "Retargeting has reached a level of precision that is leaving consumers with the palpable feeling that they are being watched as they roam the virtual aisles of online stores." and "It illustrates that there is a commercial surveillance system in place online that is sweeping in scope and raises privacy and civil liberties issues".

The natural reaction from the user community is to block trackers. However, tracker blocking approaches often cause negative impacts on the business model of the Internet, which is mainly fostered by advertising revenue (Section 1.4). Alternatively, researchers have proposed technical designs that enable targeted advertising without the need of tracking users [39][40]. Their common idea is to shift advertisers' ad personalizing algorithms to users' devices, thus providing users with complete control over their data. However, it is unclear in these approaches how the confidentiality of these algorithms is guaranteed against users. In addition, these proposals do not consider the *Real-Time Bidding* (RTB) protocol, which allows trading ad spaces at real-time auctions on a per-ad-impression basis, in their design. RTB is actually a major channel for retargeters to buy ad

spaces.

6.1.2 Our Proposal

We propose the first retargeting system which does not require tracking. In this system, a client software operating at the user's terminal collects products on websites that the user has visited, and stores them locally. The product selection algorithm is distributed between the client and the retargeter, while an effective homomorphic scheme protects the algorithm's confidentiality from the client. Specifically, the homomorphic encryption scheme allows the client to perform some precomputation for the retargeter without learning how exactly the retargeter selects the advertised products; these precomputation results are then sent to the retargeter in a way that does not leak any user private information (e.g., IP address); and finally, the retargeter selects the final products to advertise to the user. Our scheme is compatible with today's advertising systems, and integrates RTB as part of its design.

The proposed scheme has several benefits over the existing retargeting scheme. (1) It improves *user privacy* by preventing systematic tracking. (2) It provides *more transparency and control* for users over their data which is used for retargeting. In particular, users can filter out privacy-sensitive products from advertising. (3) It is *more efficient* than existing schemes since profiling is performed locally, and is therefore based on higher quality data. Furthermore, in existing systems, users frequently receive retargeting ads about products that are not relevant anymore (e.g., they already bought them from another seller). This is annoying to most users, yet inevitable, since advertisers do not always know whether users have changed their intents or made a purchase. This problem is trivially solved in our scheme since users can filter-out products that they are no longer interested in. (4) It does not rely on cookie and tracking technologies, and therefore works even if users use anti-tracking tools.

6.2 Background: Retargeting and Privacy

Before retargeting became prevalent, conventional targeted ad systems (e.g., Google Adsense) were often only interest-based: ad networks collected urls of sites visited by users (e.g., hotels.com) to infer user interest categories (e.g., traveling) and used this information to deliver personalized ads.

By contrast, *retargeting* is much more effective (and also privacy-invasive): retargeting trackers do not only attempt to identify user interests, but also aim to get the *exact products* on each page, possibly with additional information such as related user actions (e.g., search) and the point where the user suspended the purchasing process. For instance, given

a user visiting hotels.com, retargeters might learn that he is interested in a hotel in a certain district in Paris. In addition, they can also infer whether the user only looks at this hotel, or has a clear booking intent (the hotel is in a shopping cart), or already made a booking (the hotel is in a booking confirmation page).

6.2.1 Retargeting Mechanism

There are five entities in a retargeting system: *advertiser*, *publisher*, *ad exchange*, *retargeter* and *user*. Advertisers (e.g., hotels.fr) wish to promote their products by showing ads to users. Publishers (e.g., nytimes.com) develop web pages providing content to users and sell ad spaces on their pages to advertisers. Ad Exchanges (e.g., DoubleClick) connect ad buyers (advertisers or their representatives) and sellers (publishers) in real-time transactions. Retargeters (e.g., Criteo) provide retargeting service to advertisers.

We illustrate how retargeting works with the following example. A user visits maty.com, searches for an engagement ring (*product*), looks at its details and then leaves the website without a purchase. Later on, he visits accuweather.com and finds this engagement ring advertised to him on the page. In this scenario, the owner of maty.com (*advertiser*) uses the retargeting service provided by a *retargeter*, say Criteo, to retarget the user on *publisher* pages (accuweather.com in this example). This retargeting process has two main phases: tracking (the retargeter tracks users on maty.com) and delivering ads (the retargeter delivers ads to users on accuweather.com). In what follows, we elaborate these two phases.

Tracking: The advertiser puts the retargeter's tracking code on its pages and encodes in each page the ids of products aimed for retargeting (e.g., an engagement ring's id: “ring123"). When a user visits such a page, the tracking code sends his cookie (belonging to the retargeter's domain) along with these product ids to the retargeter. In addition, the advertiser can flexibly send additional user information to the retargeter. For example, if the user performs a search, the keyword and product ids on the resulting page could be sent to the retargeter. In case the user visits his shopping cart, the quantities of products in the cart could also be sent along with the product ids. This information serves a more accurate targeting (e.g., targeting shopping cart abandoners). As a result of tracking, the retargeter builds a list of potentially advertised products assigned to each user cookie.

Delivering ads: The retargeter finds ad spaces on publisher pages (e.g., accuweather.com) mostly through the *Real-Time Bidding* (RTB) protocol. RTB is provided by an Ad Exchange (ADX): publishers put the ADX's advertising code on their web pages; when a user visits such a page, the code sends an ad request, which contains the ADX's user cookie and the information about the page, to the ADX. The ADX subsequently broadcasts these data

in form of bid requests to its registered bidders, including retargeters, for them to compete in a real-time auction for the ad space.

Upon receiving a bid request, each bidder recognizes its own user cookie from the ADX's cookie thanks to a cookie matching protocol [154]. Given the list of products previously assigned to the user cookie, each retargeter selects some products[1] to be advertised to the user as well as the advertising price it would pay. This selection is typically based on the products, the user profile, and the page which displays ads.

ADX is often operated by giant firms, such as Google, Yahoo or Microsoft: they manage huge online ad inventories and put them into real-time auctions in order to maximize the revenue.

6.2.2 Privacy Concerns

There are serious privacy concerns related to the fact that products, which users are interested in, are centrally collected by retargeters. Specifically, certain kinds of products might immediately expose sensitive information: an engagement ring reveals that the user likely intends to get married while its price implies his financial capacity, a hotel booking reveals the user's destination, a bank loan reveals the user's financial status, and so forth. In addition, research has shown that users' habits can be characterized by the products they consume for a period, and that changing habits anticipate special events in their life (e.g., being pregnant, getting divorced or graduating) [58]. The inferred information can be used by marketers to better suggest products to users, but can also enable price or service discrimination [155]. Moreover, retargeting ads might reveal users' private actions, e.g., to their family members. This has been shown in a practical case when a commercial coupon advertised to a father revealed that his teenage daughter is pregnant [58]. Furthermore, all this private information can be easily linked with user identity, as commercial websites often incentivize users to provide their name, email address or telephone number, e.g., through a fidelity program.

6.3 Goals and Assumptions

6.3.1 Goals

We introduce a novel retargeting system which preserves *user privacy* from retargeters. Specifically, our scheme ensures that retargeters cannot associate any user information with user personal identity such as IP address (*anonymity*) and cannot associate multiple pieces of such information with the same user (*unlinkability*). We also prevent any man-in-the-middle attack (possibly mounted by Ad Exchanges) between users and retargeters

[1] The retargeter can use the exact products that the user visited and/or suggest relevant products.

that could aim at eavesdropping the transmitted data (e.g., ads) to learn private information (e.g., list of products visited by the user).

In addition, our scheme also protects the secrecy of retargeters which do not want to reveal every detail of their ad selection algorithm even if it is distributed between the user and the retargeter. This is a challenging task since the algorithm needs private user attributes (e.g., age, sex, or interest categories) and also confidential data from the retargeter (e.g., the combination of user attributes that yields higher clicking rate) as input.

While ensuring privacy, we also aim to keep the retargeting *effectiveness* of today's systems. Specifically, we provide retargeters with almost the same input and flexibility for ad selection as they have today. Nevertheless, we cannot compare the performance of our proposal with that of the current retargeting system due to the lack of real data and details of today's ad selection algorithms. Hence, we leave this performance comparison for future work, and rather discuss several possible improvements of our scheme.

6.3.2 Security Assumption

Some related work [39][40] assume honest-but-curious parties, which abide by the protocol rules but may misuse any information obtained in the protocol run. In our work, we assume that the retargeter and the ad exchange can be *active*, i.e., they can also mount active attacks to break anonymity or unlinkability.

However, we assume that the ad exchange does not collude with the retargeter. The major goal of an ad exchange is to provide a fair market for trading ads, not to break user privacy at all cost. Moreover, ad exchanges are often large companies with reputation (e.g., Google or Facebook), which unlikely take the risk to collude with an external party. In addition, most of their privacy policies actually contain non-collusion terms making them subject to legal action. Entities which do not have such privacy statement could be excluded by the client software by maintaining a blacklist of them.

On the client side, we assume that the user trusts the client software. The client software can be open-source (e.g., a browser plugin), and therefore can be easily audited by a trusted party. Finally, the client can be malicious towards retargeters. For example, a retargeter's competitor might manipulate the local configuration to learn the private algorithm of the retargeter.

6.4 System Overview

We follow a distributed (in contrast to the currently centralized) approach where a software agent running at the user's device, called *client*, creates and maintains user profile, as well as computes a *score* for each visited product. These scores allow retargeters to select ads

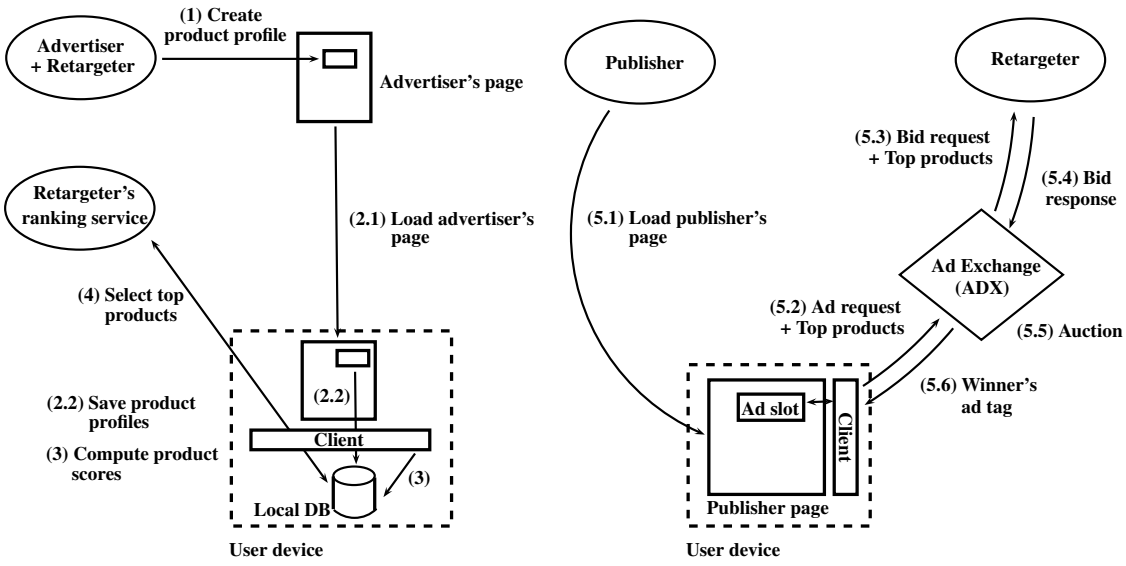


Figure 6.1: System overview

and to adjust their advertising prices to the user. At a high level, the protocol works as follows (Figure 6.1)[2]:

1. The retargeting advertiser builds product feeds for the retargeter in which it specifies the products to be advertised. The advertiser and the retargeter agree on a range of advertising details for each product such as product quality (e.g., inferred from the number of users showing interest in the product), user targeting criteria and ad pricing[3]. These details are encoded for each product as a *product profile* and embedded into the advertiser's web pages.

2. When a user visits a commercial web page (which belongs to a retargeting advertiser, e.g. hotels.com) looking for a product, the client retrieves the product profile from the page.

3. The client then computes the score of this product by matching the product profile with the user profile. This score gives an estimation of the expected revenue of the retargeter from advertising this product to the user, and therefore shows whether this product is a good candidate for retargeting or not.

4. The client selects the products having the highest scores, called *top-products*, for each retargeter that it encountered.

5. When the user visits a publisher's page (e.g., accuweather.com), which contains an advertising frame, he sends the top m (say $m = 3$) product ids and scores of each

[2]Note that most values that are processed in the protocol are encrypted. We ignore this aspect at this stage for simplicity.

[3]These details can be set by advertisers or suggested by retargeters.

retargeter to the ADX[4]. The ADX initiates the RTB auction among all retargeters by distributing the top-products of each retargeter. Based on the scores of the user's top products, each retargeter decides whether to serve an ad to the user and at what price, and sends the bid to the ADX. Finally, the ADX puts the winner's ad creative on the publisher's web page; the ad creative anonymously loads ads from the retargeter through a proxy mechanism at the ADX.

As product profiles are considered as commercially sensitive information, they are encrypted by retargeters using a homomorphic encryption scheme such that users cannot access the profile attributes. In order to select the highest scored products, the client invokes the *ranking service* of each retargeter. In particular, the client sends the list of *encrypted* scores to the ranking service in a way (described later) that leaks neither the list of products nor the user profile to the retargeter. The ranking service decrypts the scores, sorts them, and sends back the sorted list of encrypted scores to the user.

The ADX *does not* include any user cookie (or user identifying information) into bid requests as in today's RTB protocol. Meanwhile, the top products sent to retargeters are encrypted and therefore inaccessible to the ADX. Retargeting ads are loaded through a proxy mechanism at the ADX and similarly are encrypted in order to prevent the ADX from learning their content. Note that our scheme requires that retargeters buy ad spaces from publishers only through ADX[5].

6.5 System Details

6.5.1 Product Score Evaluation

We present our approach to compute product scores in the Cost-Per-Click (CPC) model, i.e., advertisers only pay retargeters when users click on ads. A similar approach can be applied to other models such as Cost-Per-Mile or Cost-Per-Action. In CPC, the retargeter's expected revenue from advertising a product P to a user U is calculated as $CTR_U(P) \times CPC(P)$, where $CTR_U(P)$ is the estimated Click-Through-Rate (CTR) of the ad shown to this user, and $CPC(P)$ is the price that the advertiser pays for an ad click. The product score in our scheme is defined as this expected revenue.

We assume that each product P has a default CTR, denoted by $CTR(P)$, and computed by a retargeter R, e.g., based on the click history of similar products. A product initial

[4]We assume a mechanism such that retargeters notify users of which ADXs they are currently working with (e.g. by putting this information on a website accessible to clients). The client only includes products of the related retargeters when it detects an ADX.

[5]We believe this requirement is reasonable: major retargeters (e.g. AdRoll, Criteo) are actually partners with major ad exchanges (e.g. DoubleClick, RightMedia, Facebooke, etc.) for their *indirect buying* through RTB [156][157][158], and start embracing Preferred Deals as an efficient technique for their *direct buying* relationships with publishers [159][160][161][162].

score (PIS) of P is calculated as $PIS = CTR(P) \times CPC(P)$, which is independent of users. R targets users based on a set of *user attributes* such as $\{gender, age, interests, location\}$. Example values of these attributes are {"*male*", "$24-35$", "*sport*", "*Paris*"}. For each of these attribute values, such as *male*, R quantifies its effect on the CTR of P by an *impact factor*, such as 1.2, indicating that advertising P to a *male* user would increase P's CTR by 20%. Impact factors can be learned from statistics, e.g., by analyzing the CTR of similar products when being advertised to *male*. The impact factor which is larger/smaller than 1 increases/decreases P's CTR. If the retargeter does not have sufficient statistics to measure an impact factor, it sets that to a default value (e.g., 1).

The retargeter R configures, for each product, the initial score PIS and the impact factors for all possible values of each user attribute. These properties are then encrypted by R's symmetric key using a homomorphic encryption scheme such that a user can select the encrypted impact factors corresponding to his profile (e.g., a *male* user picks the encrypted impact factor for *male*) without knowing their values. The user performs this selection and then computes the product score using the (encrypted) PIS and the selected (encrypted) impact factors. In the following, we describe the details of our approach.

6.5.1.1 Encoding User and Product Profiles.

The user attributes in our scheme include *age*, *gender*, *location* and *interest categories*. In addition, they can also include three extra attributes that are different for each product: *user conversion status* (e.g., the product was put into shopping cart but not purchased), *frequency of visits* (e.g., visits per day) and *time of last visit* (e.g., last hour or last day)[6]. Note that this list is not exhaustive: additional attributes can be added depending on specific targeting purposes.

- **User profile:** A user profile is described by a vector U where each coordinate encodes the value of one user attribute. Specifically, each coordinate is the index of the value of the corresponding attribute in its value set. An example of a user profile is $U = (1, 0, 89, 1, 2, 2, 3)$ where each coordinate may encode $(1 :$ "$18-24$"$(age), 0 :$ "*male*"$(gender), 89 :$ "*Madrid*"$(location), 1 :$ "*computer science*"$(interest), 2 :$ "*in shopping cart*"$(co$ "*10 visits/day*"$(frequency), 3 :$ "*last week*"$(last\ visit))$.

- **Product profile:** A product profile belonging to a retargeter R includes (1) the ids of P and R (id_P and id_R, resp.); (2) PIS; (3) a url of R's ranking service (see later); (4) a set of vectors $\{F_1, ..., F_n\}$, where each corresponds to a user attribute and contains impact factors for all possible values of that attribute. Given the previous example,

[6]Users who put the product into shopping cart are more likely to make a purchase than those only looking at the product. Larger visit frequency usually anticipates a purchase. Similarly, recently visited products are more likely to be purchased.

F_2 corresponds to *gender*, while $F_2[1]$ and $F_2[2]$ are the impact factors of *male* and $female$, respectively. [7]

6.5.1.2 Computing Score.

For each attribute i, the client uses $U[i]$ to pick one impact factor from F_i, namely $F_i[U[i]]$. The score S_P^R for product P, which belongs to a retargeter R, is computed as follows:

$$S_P^R = PIS_P^R \times \prod_{i=1}^{n} F_i[U[i]] \tag{6.1}$$

Although this formula can be computed using a *multiplicative* homomorphic encryption scheme[8], such as El-Gammal [163], such asymmetric encryption is too costly in practice. We, instead, use an *additive* homomorphic encryption scheme based on *symmetric keys*, which is proposed in [164] (Appendix A), for its efficiency. Hence, we convert Formula 6.1 to additive form by taking the logarithm of both sides and use the resulting formula with the additive scheme:

$$\log S_P^R = \log PIS_P^R + \sum_{i=1}^{n} \log F_i[U[i]] \tag{6.2}$$

For all i, R encrypts the logarithm of F_i such that each coordinate of F_i is encrypted with a different key. Specifically, for any F_i, R computes $F_i^E = (Enc_{k_{i,1}}(\log F_i[1]), \ldots, Enc_{k_{i,|F_i|}}(\log F_i[|F_i|]))$ where $k_{i,j} = Hash(id_P|K|i|j)$ and K is the secret key of R. In addition, R computes $Enc_{k_{PIS}}(\log PIS_P^R)$, where $k_{PIS} = Hash(id_P|K|\text{“}PIS\text{”})$.

To compute the product score, the user simply applies Formula 6.2 but in the encrypted domain, i.e.,

$$Enc_{k_{PIS}}(\log PIS_P^R) + \sum_{i=1}^{n} F_i^E[U[i]] = Enc_{k_{PIS}+\sum_{i=1}^{n} k_{i,U[i]}}(\log S_P^R) \tag{6.3}$$

where the equality follows from the homomorphic property of *Enc* [164]. The product score can be retrieved by taking the exponent of the decrypted value.

6.5.2 Product Ranking

The client needs to rank products belonging to the related retargeter R when it encounters a new product profile. Alternatively, it can perform the ranking periodically (e.g., hourly)

[7] Note that all values of the impact factors are converted to integer by using the micro format (e.g., 1 is converted to 1,000,000 micros). The micro format is commonly used in the advertising industry.

[8] An encryption scheme, denote by *Enc*, is additive homomorphic if, given two arbitrary plaintexts m_1 and m_2, it allows computing $Enc(m_1 + m_2)$ from $Enc(m_1)$ and $Enc(m_2)$ without decrypting any of these values. Similarly, the scheme is multiplicative homomorphic if it allows computing $Enc(m_1 * m_2)$ from the ciphertexts $Enc(m_1)$ and $Enc(m_2)$.

in case new products are frequently recorded. Recall that the URL of the ranking service is included in each product profile of R.

To rank a set of products, the client sends the list of product scores, computed in Formula 6.3, to the ranking service of R. However, in order to decrypt this score, R would need the set of keys $\sum_i k_{i,U[i]}$, which eventually reveals the user profile U. We instead follow a popular approach [41][165] and propose to implement the whole ranking procedure using a secure co-processor (SC) (e.g., IBM 4765 [166]), which provides secure storage as well as trustworthy, programmable execution environment. The SC could be deployed at R, and users can verify the code being executed on the SC through a remote code attestation mechanism (e.g., [167]). As the SC is tamper-resistant, while the communication between the client and the SC is encrypted, no other parties (including R) will learn anything about user profiles and top-products.

The ranking algorithm is simple and can be public. In particular, R installs K and the public ranking procedure on the SC. Afterwards, the client establishes a secure connection to the SC (e.g., through TLS), and sends U, as well as id_P with $Enc_{k_{PIS}+\sum_i k_{i,U[i]}}(\log S_P^R)$ for the related products to the SC. Then, the SC can decrypt product scores, rank these products, and send back the sorted list of product ids to the client.

6.5.3 Ad Serving

The client sends ad requests, which include the top products' ids, id_Rs (retargeter ids), PIS and scores, to the ADX. Note that, as opposed to what in current RTB systems, ad requests in our scheme do not contain any cookie. The ADX uses id_Rs to separate the top-products for each retargeter and includes them along with the user's visiting page in related bid requests sent to appropriate retargeters. Each retargeter R decrypts the product scores and adjusts these scores according to the quality and relevance of the visiting page. If there is a difference between the PIS of a product and the latest PIS at the retargeter (e.g. due to a CPC change), it updates the product score accordingly. R then selects the product with highest score and determine the bid price based on the score. Subsequently, R builds an ad creative which contains the ad url to load ad for this product from its server, and then includes the creative and the bid price in a bid response which is sent to the ADX.

If R wins the auction, its ad creative is sent to the user's device. Note that current ADXs mostly allow these creatives to load ads directly from the retargeter's server or from the ADX' storage which contains retargeters' pre-uploaded ads. Both approaches do not protect user privacy: while the former exposes user IP address to the retargeter, the latter reveals personalized ad content to the ADX. In our scheme, we protect user privacy by requiring that ads be encrypted by the retargeter and loaded through a proxy mechanism at the ADX. The resulting computational and bandwidth overhead is analyzed in Section 6.7.

In particular, the ADX replaces the ad url in R's ad creative with a url pointing to the

ADX which contains the original url as value of a HTTP parameter. At the user's device, the creative requests the ad from the ADX; the ADX retrieves the retargeter's ad url, loads the ad from the retargeter, and then returns the ad to the user. The ad view or click report is sent to the ADX, which subsequently removes any user related information (e.g. IP address, browser info, etc.) and forwards the report to the retargeter.

Protecting Ad Requests and Content: The product ids (in ad requests) or the ad content could reveal the user's top products to the ADX. In order to protect these data, we use a symmetric key which is shared between the retargeter and the client. In particular, before sending an ad request, the client generates a session key $K_{u,R}$ for each retargeter R whose products are selected for the request. The client then encrypts each session key with the public key of the corresponding retargeter and includes the resulting encrypted key in the ad request. The ADX subsequently includes the encrypted keys in bid requests. R first decrypts $K_{u,R}$ with its private key and then decrypts the top-products' ids with $K_{u,R}$. $K_{u,R}$ is also used by R to encrypt the ad content for the client if R wins the auction.

6.5.4 Other Features

Frequency Capping: Frequency capping restricts the number of times an ad is shown to the same user (e.g., less than 10 times per day). This is solved trivially in our scheme: the client counts the number of times a product is advertised to the user, and ignores a product when building ad requests if its counter is beyond a threshold.

Click Fraud Defense: Since the click report is anonymized (by the proxy mechanism at the ADX), this makes the click fraud more difficult to detect. However, a similar approach to [39] could be applied: the retargeter sends suspected click reports to the ADX; the ADX traces back to the user IPs which are responsible to these clicks to examine the probability of click fraud.

6.6 Privacy Analysis

In this section, we analyze how user privacy is protected from retargeters, ad exchanges, advertisers, and other users. We also analyze how the confidentiality of the retargeter's ad selection algorithm is guaranteed.

6.6.1 Retargeter

A retargeter R might get user information from bid requests or ranking requests. Although R gets top-products from bid requests, it cannot associate them with user identifying

information such as the IP address (which is not included in bid requests). Contrarily, while *R* can see the user IP from ranking requests, it is not able to get any related user data since these requests are received and processed by the SC through a secure connection. The tamper-proof property of SC prevents *R* from intercepting and learning any internal data during the ranking process. In summary, *R unlinkably* gets users' top products (from bid requests) and users' IP addresses (from ranking requests).

Without covert channel or collusion, *R* cannot break *user anonymity* unless it can correlate ranking requests with bid requests, e.g., if there are too few users, or ranking request time correlates with bid request time. Although both attacks seem impractical with large number of users (which is likely the case), the time correlation attack can be further mitigated by randomizing the time of ranking requests.

Nevertheless, *R* might attempt to link users' top products from different bid requests and gradually build unique profiles of users (*linkability*). This would be difficult given that the client sends only a small number of top-products in ad requests. A possible attack is to infer user attribute values from the product scores and then use them as a fingerprint to link different top products. This, however, can be mitigated by coarsening score values at ranking (performed inside the SC) and then using these coarsened values in ad requests.

Although *R* can hardly break user anonymity or profile unlinkability if it faithfully follows the protocol, it might collude with other parties in order to do that. Recall that we assume non-collusion between ADX and *R* (Section 6.3). Other parties that might (and in fact have motivation to) collude with *R* are the advertiser and the publisher.

Assume that *R* colludes with a malicious advertiser. For example, the advertiser might send all its log entry database (possibly containing visited products, time of visits, and the user IP address of every visit) to *R*. In order to link bid requests, which likely contain products from other advertisers, with IP addresses from this database, *R* may perform timing or product frequency analysis.

- *Timing analysis: R* correlates the visiting time related to a product (in the malicious advertiser's database) with the reception time of a bid request (e.g., if they are close, both events are possibly originated from the same IP). However, as discussed previously, time-based correlation is difficult due to the large number of users. In addition, it is hard to predict the interval between a visit to an advertiser's page and a visit to a publisher's page.

- *Product frequency analysis: R* selects products from the database which were visited by a small number of users, ideally only by a single user (identified by IP address). If a bid request contains any of these products, *R* can associate it with a small group of users, or a single user, accordingly. Though this attack is possible, it can only affect a very small proportion of users. In addition, in this form of attack, *R* cannot actively target a victim.

The timing attack would be more practical if *R* colludes with a malicious publisher, as the time of a visit to a publisher page is very close to the time of the resulting bid request. However, this only works in case of very small publishers which have few connecting users at a time. In addition, the client can add some arbitrary delays in sending bid requests in order to mitigate the risk of such attack.

6.6.2 Ad Exchange

Although the ADX can see the user's IP address, it cannot obtain the list of top-products or ad content; they are encrypted using the session key shared between the user and the retargeter. The ADX cannot forge fake session keys in order to decrypt retargeting ads served through its proxy mechanism. In particular, if a fake session key is produced, the retargeter cannot get the right product ids from the related request.

Notice that, as part of the RTB protocol, ADXs can still get urls of visited sites in our solution. Nevertheless, the risk of profiling users using these urls is less severe than that in retargeting (Section 6.3). Moreover, users can block cookies to prevent possible tracking performed by the ADX; our scheme works without the need of any tracking cookie. In this work, we focus on the privacy risks of retargeting, and leave the total elimination of this url leakage towards the ADX for our future work.

6.6.3 Advertiser

In our system, an ad click brings the user directly to the advertiser's site, the same as what is happening today. We acknowledge that this is a problem since advertisers can leverage fine-grained targeting feature and high user profiling quality in our scheme to learn more information about users than what they can learn in today's system. One solution could be to handle post-click sessions through an anonymized network, such as TOR [168]. This might, however, lead to additional network latency, complexity in measuring ad performance or other implication, and therefore need deeper analysis from the research community and the advertising industry.

6.6.4 User

Targeted ads were proved to be a potential source of leaking user private information due to its personalized content [169]. In case of retargeting ads, someone happens to look at a user's screen when he is browsing the web in a public place (e.g. at work) may infer his previous private actions (e.g. at home)[9] that the user may want to keep secret (e.g. looking for an engagement ring). Since user profiles in our scheme are built and stored locally,

[9]We assume that the user is using the same laptop computer in both environments.

users can trivially filter out sensitive products to prevent such unexpected information leakage. The encryption of transmitted data also prevents eavesdroppers from exploiting personalized ad content thereby inferring user private information.

The client can be malicious toward a retargeter *R*. For example, one of *R*'s competitors might install and manipulate a client in order to learn product profiles of *R*. Since the used encryption scheme is proved to be *perfectly secure* [164] (Appendix A), only *R* can decrypt its product profiles. Nevertheless, a malicious client may attempt to manipulate its own user profile to observe possible changes in the ranking list and thereby inferring some properties of the product profiles. This kind of attack needs to be repeated many times in order to learn meaningful results. Consequently, it can be mitigated by applying a threshold on the number of ranking requests per client. Furthermore, the ranking service on the SC may apply a randomization in the order of top-products to make such kind of attacks more difficult, if not impossible.

6.7 Implementation and Evaluation

6.7.1 Implementation

To build the client, we extend the Firefox plug-in *HttpFox* [141] and use SQLite for storing local data. The Ad Exchange and the Retargeter are written using NodeJS [170]. Note that the ranking service, which is supposed to be implemented on a SC, is implemented in NodeJS and executed by a normal processor in our implementation. We ran our client inside a Firefox browser on a laptop running OS X 10.7.2 on an Intel Core i5 2.4 GHz, and the retargeter and ad exchange on a machine with an Intel Core 2 Duo 2.66 GHz and running Ubuntu 11.04.

Based on Google Ads Settings [171], we configure the system with 2 genders, 7 age ranges, 24 top interests and 846 word localities. The user conversion status, the frequency of visits, and the time of last visit are all configured with 5 permissive values. The client stores up to 1000 products (for all retargeters) and includes 3 products per retargeter in each ad request.

6.7.2 Evaluation

We evaluate, in this section, the computational and bandwidth overhead of our scheme (compared to the existing system) through an example scenario.

Example scenario: A retargeter *R* provides retargeting services for 100 advertisers, each having 1000 retargeting products. Each day there are 1 million unique users browsing these advertisers' websites (10K users per site on average). Each user browses for 20

Table 6.1: Computational overhead at retargeter (per day)

Index	Action	Time (hours)
1	Encrypting 100K products on advertisers' websites	0.21
2	Decrypting 60M scores in bid requests	0.83
3	Decrypting 2M session keys	1.23
4	Encrypting 2M ads with session keys	0.46
5	Processing 20M ranking requests	0.33
	Total	3.06

retargeting products which belong to 3 retargeters on average, and issues 20 ad requests containing such products, per day. Ad requests are handled by a single Ad Exchange (ADX), resulting in 20 millions requests per day (about 232 requests per second) received by the ADX. For each ad request, the ADX sends bid requests to all retargeters whose products are contained in the ad request. The retargeter R submits bid responses for all 20 millions bid requests, wins about 10% of these auctions (2 millions winning times).

6.7.2.1 Computational Overhead.

Our client can perform 5K homomorphic computations per second. It can generate 200 session keys (encrypted by *R*'s public key) and 30 ad encryptions per second. With a few dozen score computations and ad requests per day, the computation at client causes a negligible overhead.

Our retargeter can perform 133 product encryptions, 100K score decryptions, and 6K ad encryptions per second. The computational overhead of the retargeter (per day) is shown in Table 6.1. In this table, (1)(the first line) is estimated in the worse case (the retargeter re-encrypts all 100K products everyday), while (3) can be significantly optimized by offloading asymmetric operations using dedicated hardware [172]. With our implementation, assuming that the retargeter rents computation resources from Amazon EC2 [173] (e.g., using a c3.large instance which is optimized for computation purposes), the total computational overhead costs about $0.45 per day.

6.7.2.2 Bandwidth and Storage Overhead.

Each product profile increases the size of the product web page by 6 KB. This is a negligible overhead given the fact that, for example, a maty.com's product page's HTML source is around 100 KB, excluding images, css and JavaScript files. The sizes of an ad request and a ranking request are 23.94 KB and 15.96 KB, respectively. The total bandwidth overhead for each user is therefore approximately 2 MB per day. The products stored at a user's device (maximum 1K) cause a maximum 8MB local storage.

Table 6.2: Bandwidth overhead at ADX and retargeter (per day)

	ADX		**Retargeter**	
Index	Action	Bwth	Action	Bwth
1	Receiving 20M ad requests	53GB	Receiving 20M bid requests	18GB
2	Sending 60M bid requests	53GB	Receiving 20M ranking requests	35GB
3	Proxying 10M retargeting ads	190GB		

The bandwidth overhead at the ADX and the retargeter are presented in Table 6.2. In order to provide an estimation of the resulting cost, we assume that both the retargeter and the ADX run their software on Amazon EC2's servers. Amazon EC2 only charges the bandwidth from EC2 to the Internet, which is related to ads served by the ADX to users. If we apply the upper bound of this price, namely \$0.12 per GB, the bandwidth overhead resulting from serving ads (190 GB) costs the ADX approximately \$22.8 per day ($\$2.28 * 10^{-6}$ per ad). This additional cost can be shared among retargeters, advertisers, and publishers, for example by the ADX slightly increasing the transaction commission. Given the significant number of advertisers and publishers working with an ADX in general, the cost per entity would become negligible.

6.8 Discussion

6.8.1 Compatibility

Our purpose is not to replace but rather complement the current retargeting system by providing an alternative solution for retargeters (and ad exchanges) to provide retargeting ad service in a privacy-preserving manner. An example scenario would be as follows. The retargeters choose privacy-preserving (our scheme) or regular (current scheme) RTB mode at the ADX. At auction, the ADX sends privacy-preserving retargeting bid requests (as described in our scheme) to privacy-advocate retargeters, and regular bid requests to the others. The ADXs which support our scheme specify this in their ad requests (e.g., by using a special HTTP header, such as "*PPRetargeting = true*"). The client[10] intercepts and includes into these requests the respective user's top retargeting products. If users do not favor traditional retargeting or RTB, they can, for example, configure the web browser to disable third-party cookies.

[10]Privacy-advocate retargeters and ADXs encourage users to use the client software.

6.8.2 Scoring Algorithm

The score computation (i.e., $CPC(P) \times CTR(P) \times \prod_{i=1}^{n} F_i$), is based on an assumption that the effect of user attributes on the product's CTR are independent of each other. Consequently, this algorithm has a limitation: it does not take into account the intrinsic correlation between user attributes. For example, say a user's age "18-24" and interest "sport", each increasing a product's CTR with 1.2 (20%) and 1.1 (10%), respectively, a combination of them might not necessarily be equal to $1.2 \times 1.1 = 1.32$, but can be higher or lower than that depending on the correlation between the two attributes. In the following, we discuss a possible extension that takes into account this correlation.

Attribute Coefficients: We assume that the correlation between two attributes F_i (e.g., age) and F_j (e.g., gender) can be quantified by a coefficient C_{ij}, so that their combined effect on a product's CTR is computed as $F_i \times F_j \times C_{ij}$. Each C_{ij} can be defined with different values for different value ranges of F_i and F_j. The score in this case would be computed as: $CPC(P) \times CTR(P) \times \prod_{i=1}^{n} F_i \times \prod_{i,j} C_{ij}$. Note that the correlation coefficient can be computed for more than two attributes, e.g., C_{ijk} or C_{ijkl}. In the worse case, the size of the coefficient set is equal to the number of all possible combinations of attribute values. The coefficient values are also encrypted by the retargeter, and their applied attribute ranges can be defined in a script encoded into the product profile.

6.8.3 Gathering Statistics

User statistics (e.g., click behaviors of a group of users) are important for retargeters to enhance their retargeting performance. In our scheme, the SC can also be used to aggregate this information in a privacy-preserving manner. In particular, after an interval (e.g., a week), the client sends the user's profile and CTRs of local products to the SC (through a secure connection). The SC aggregates these data from a sufficient number of users, produces statistical results (e.g., average CTR of a product advertised to *sport-enthusiastic* users), and sends them to the retargeter.

6.9 Summary

Retargeting ads are growingly rampant and cause great privacy concerns, mostly resulting from tracking user intents. In this work, we propose the first retargeting system that does not rely on tracking. Our scheme leverages homomorphic encryption to distribute the ad selection algorithm between the user and the retargeter in a way that securely combines confidential data from both parties. The proposed scheme is compatible with RTB and

supports major characteristics of current ad systems such as real-time auction, fine-grained targeting, ads freshness and frequency capping.

This is, to our knowledge, the first work that considers RTB in a privacy-preserving advertising solution. We note that RTB is increasingly prevalent and plays an important role in today's targeted advertising systems, itself exposing serious privacy concerns [144]. Enhancing user privacy in RTB in general is therefore our goal in a near future.

Chapter 7

Conclusion and Future Directions

Privacy in targeted advertising has long been a source of controversy and debates. On one hand, targeted advertising brings tremendous economic benefits: it provides a means for advertisers to optimize their advertising resources and performance, increases advertising revenue for publishers, and serves more useful ads to users. On the other hand, it invades user privacy as third-party companies are collecting vast amount of users' activities - most of these data are not voluntarily shared by users. Resolving privacy problems poses a multidisciplinary challenge which requires participation of economists, the research community and the industry. The major roles of the research community are to study potential privacy risks in current systems and, more importantly, to work out privacy friendly solutions which are economically viable.

In this thesis, we studied possible privacy leakages in current systems through targeted ads delivery and ad exchange protocols. We showed that, while related entities claim not to disclose user private information to any external parties, decent protection of these data is not guaranteed. In addition, we proposed a novel design for retargeting advertising without the need of tracking, which can be integrated in the current ad exchange model and compatible with the existing advertising infrastructure.

Solving privacy problems is beneficial not only to users but also to advertising companies. We expect that the privacy leaks exposed in this thesis could help companies in anticipating privacy problems and finding adequate measures, while the proposed solution contributes to the debates in reconciling the two conflicting goals of targeting and privacy. There are various ways to extend or complement our work towards a privacy-friendly targeted ad system. In the following, we present possible extensions to our work and other future directions.

Extensions to our work:

- **Targeted ads detection:** Our targeted ads filtering technique is probably not optimal, and can be improved in many ways. Such technique can be used to develop a client tool to detect targeted ads, which can be helpful in detecting tracking behaviors

which are not visible from the client side. For example, privacy watchdogs can use this tool to check whether a company performs stateless tracking, such as passive fingerprinting. Another application of the tool could be checking companies' compliance to Do-Not-Track, that is, if they still collect user information given the presence of the Do-Not-Track signal.

- **Cookie re-spawning using cookie matching:** Since trackers can map their cookies with those from external parties thanks to cookie matching, they can potentially link their different cookies of the same user. For example, when users delete these trackers' cookies, trackers can possibly leverage external cookies to re-spawn the deleted cookies or assign users with new cookies which are linkable to the deleted ones. Investigating the existence or prevalence of such actions would be an interesting future study to complement our work.

- **Privacy-preserving Real-Time Bidding (RTB):** While the solution proposed in this thesis is aimed at retargeting over RTB, an interesting future direction would be to extend this model to other kinds of targeted ads over RTB. There are actual demands for that. For example, as for privacy concerns, Facebook ad exchange currently does not share with real-time bidders any information about the user's context, i.e. the page he is browsing on Facebook[1]. A privacy-preserving RTB protocol, if implemented, could bring benefits to advertisers by giving them access to such information in order to increase ad targeting performance.

Other future directions:

- **Tracking prevention:** Most of privacy-enhancing tools provide users with lists of trackers and options to block/unblock each tracker in the lists. Unfortunately, users often cannot distinguish among these trackers in order to make meaningful blocking decisions. To remedy this, an interesting study direction is to produce a tracker classifying taxonomy which is easily understandable to users. For example, trackers can be categorized based on combination of various properties, such as the sensitiveness of their collected data, their conformity to privacy regulation, their determination in tracking (e.g. use fingerprinting or super cookies), their data protection guarantee and their data usage (e.g. using data for themselves or selling them to other parties). A simple and meaningful taxonomy could help privacy tools providers in setting reasonable default tracking prevention, e.g. only block misbehaved trackers, and enhancing users' choices and knowledge, which consequently improve trackers' behaviors in long-term.

[1]This was verified during an invited talk given by the authors at Criteo's headquarter.

- **Cross-device targeting for privacy-preserving ad systems:** Privacy-preserving ad systems mostly share the idea of storing user behavioral data locally. Since targeted advertising is moving towards the trend of targeting users across different devices, it is not clear how to adapt these non-tracking approaches to such rising demands. Achieving a privacy-preserving cross-device ad targeting model remains an open challenge for future work. A possible solution could be to store user local profiles in the cloud and synchronize these data across different devices, with users' identities being their authentication data on first-party websites. However, this approach needs thorough studies on, for example, the possibility of collecting such authentication information and algorithms to avoid conflicts among users temporarily sharing a device.

- **Formalizing privacy-economics trade-offs:** The crux of resolving privacy problems in targeted advertising is to balance privacy protection with economic benefits, i.e. to reach a reasonable trade-off between these two parameters. Even though several privacy-preserving solutions have been put in perspective, there is a lack of standardized frameworks and criteria to evaluate their privacy guarantee and economic impacts (including the implementation and operating cost). Such frameworks would be necessary to assess the feasibility and compare the efficiency of different solutions.

In the Internet era, users' behaviors are increasingly being tracked, analyzed, and used for various purposes without users' knowledge or consent. Such privacy invasion is a problem that needs due attention and solutions before privacy violation becomes a fait accompli. This thesis contributes in understanding potential privacy risks and proposing a practical solution in the case of targeted advertising, which is the main motivation for tracking.

Bibliography

[1] S. Yuan, A. Z. Abidin, M. Sloan, and J. Wang, "Internet advertising: An interplay among advertisers, online publishers, ad exchanges and web users," *arXiv preprint arXiv:1206.1754*, 2012.

[2] Adage, "Affinicast unveils personalization tool." http://adage.com/article/news/affinicast-unveils-personalization-tool/2714/, 1996.

[3] C. Emert, "Web advertisers get new tool." http://www.sfgate.com/business/article/Web-Advertisers-Get-New-Tool-2983566.php, 1998.

[4] H. Beales, "The value of behavioral targeting." http://www.networkadvertising.org/pdfs/Beales_NAI_Study.pdf, 2009.

[5] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen, "How much can behavioral targeting help online advertising?," in *WWW*, 2009.

[6] W. Price, "Investment insights." http://www.rcmtechnologytrust.co.uk/Tenants/AGITrusts/Content/Documents/Posts/RTT_InvInsights301112.pdf, 2012.

[7] J. Newman, "Kerry-mccain privacy bill: What you need to know." http://www.pcworld.com/article/225039/Kerry_McCain_Privacy_Bill_What_You_Need_to_Know.html, 2011.

[8] E. Lee, "Sen. rockefeller: Get ready for a real do-not-track bill for online advertising." http://adage.com/article/digital/sen-rockefeller-ready-a-real-track-bill/227426/, 2011.

[9] FTC, "Making sure companies keep their privacy promises to consumers." http://www.ftc.gov/news-events/media-resources/protecting-consumer-privacy/enforcing-privacy-promises, 2014.

[10] EFF, "How online tracking companies know most of what you do online (and what social networks are doing to help them)." https://www.eff.org/deeplinks/2009/09/online-trackers-and-social-networks, 2009.

[11] Google, "Google's privacy policy." https://www.google.com/policies/privacy/index.html#infouse, 2014.

[12] B. Gellman, A. Soltani, and A. Peterson, "How we know the nsa had access to internal google and yahoo cloud data." http://www.washingtonpost.com/blogs/the-switch/wp/2013/11/04/how-we-know-the-nsa-had-access-to-internal-google-and-yahoo-cloud-data/, 2013.

[13] J. Harper, "It's modern trade: Web users get as much as they give." http://online.wsj.com/news/articles/SB10001424052748703748904575411530096840958, 2010.

[14] Bluekai, "Bluekai: Privacy policy." http://bluekai.com/privacypolicy.php, 2014.

[15] S. Kroft, "The data brokers: Selling your personal information." http://www.cbsnews.com/news/the-data-brokers-selling-your-personal-information/, 2014.

[16] A. Korolova, "Privacy violations using microtargeted ads: A case study," in *ICDM Workshops*, 2010.

[17] FTC, "Staff report: Public workshop on consumer privacy on the global information infrastructure." http://www.ftc.gov/reports/staff-report-public-workshop-consumer-privacy-global-information-infrastructur 1996.

[18] FTC, "Ftc announces two significant efforts in its comprehensive examination of consumer privacy." http://www.ftc.gov/news-events/press-releases/1997/03/ftc-announces-two-significant-efforts-its-comprehensive, 1997.

[19] FTC, "Ftc privacy report." http://www.ftc.gov/news-events/media-resources/protecting-consumer-privacy/ftc-privacy-report, 2014.

[20] FTC, "Ftc recommends congress require the data broker industry to be more transparent and give consumers greater control over their personal information." http://www.ftc.gov/news-events/press-releases/2014/05/ftc-recommends-congress-require-data-broker-industry-be-more, 2014.

[21] IAB, "About the iab." http://www.iab.net/about_the_iab, 2014.

[22] DAA, "Opt out from online behavioral advertising (beta)." http://www.aboutads.info/choices/, 2014.

[23] DAA, "Your adchoices." http://www.youradchoices.com/, 2014.

[24] R. Reitman, "The daa's self-regulatory principles fall far short of do not track." https://www.eff.org/deeplinks/2011/11/daa-self-regulation-principles-fall-far-short-do-not-track, 2011.

[25] J. Mayer, "A brief overview of the supplementary daa principles." http://cyberlaw.stanford.edu/node/6755, 2011.

[26] R. Gellman and P. Dixon, "Many failures: A brief history of privacy self-regulation in the united states." http://www.worldprivacyforum.org/wp-content/uploads/2011/10/WPFselfregulationhistory.pdf, 2011.

[27] J. R. Mayer and J. C. Mitchell, "Third-party web tracking: Policy and technology," in *IEEE Security and Privacy*, 2012.

[28] P. G. Leon, B. Ur, R. Balebako, L. F. Cranor, R. Shay, and Y. Wang, "Why johnny can't opt out: A usability evaluation of tools to limit online behavioral advertising," in *CHI*, 2012.

[29] J. Mayer and A. Narayanan, "Do not track: Implementations." http://donottrack.us/implementations, 2014.

[30] W3C, "Tracking protection working group." http://www.w3.org/2011/tracking-protection/, 2014.

[31] J. A. Fairfield, "Do-not-track as default," *Nw. J. Tech. & Intell. Prop.*, vol. 11, no. 2, p. 575, 2013.

[32] L. F. Cranor, "If you choose not to decide, your web browser will make your choice." http://www.techpolicy.com/Blog/June-2012/If-you-choose-not-to-decide,-your-web-browser-will.aspx, 2012.

[33] B. Szoka, "The paradox of privacy empowerment: The unintended consequences of "do not track"," in *W3C Workshop: Do Not Track and Beyond*, 2012.

[34] Eyeo GmbH, "Adblockplus - surf the web without annoying ads!." https://adblockplus.org.

[35] Ghostery, "Ghostery." https://www.ghostery.com/en/, 2014.

[36] Disconnect, "Disconnect - online privacy & security." https://disconnect.me/, 2014.

[37] Microsoft, "Tracking protection list." http://ie.microsoft.com/testdrive/Browser/TrackingProtectionLists/, 2014.

[38] A. Juels, "Targeted advertising ... and privacy too," in *CT-RSA*, 2001.

[39] S. Guha, B. Cheng, and P. Francis, "Privad: Practical privacy in online advertising," in *NSDI*, 2011.

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

[41] M. Backes, A. Kate, M. Maffei, and K. Pecina, "Obliviad: Provably secure and practical online behavioral advertising," in *S&P*, 2012.

[42] A. White, "Ip addresses are personal data, e.u. regulator says." http://www.washingtonpost.com/wp-dyn/content/article/2008/01/21/AR2008012101340.html, 2008.

[43] A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. J. Hoofnagle, "Flash cookies and privacy." http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1446862, 2009.

[44] S. Kamkar, "Evercookie – never forget." http://samy.pl/evercookie/, 2012.

[45] T. Vega, "New web code draws concern over privacy risks." http://www.nytimes.com/2010/10/11/business/media/11privacy.html?hp&_r=0, 2010.

[46] P. Eckersley, "How unique is your web browser?," in *PETS*, 2010.

[47] G. Acar, M. Juarez, N. Nikiforakis, C. Diaz, S. Gürses, F. Piessens, and B. Preneel, "Fpdetective: Dusting the web for fingerprinters," in *CCS*, 2013.

[48] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, "Cookieless monster: Exploring the ecosystem of web-based device fingerprinting," in *S&P*, 2013.

[49] F. Roesner, T. Kohno, and D. Wetherall, "Detecting and defending against third-party tracking on the web," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, NSDI'12, (Berkeley, CA, USA), pp. 12–12, USENIX Association, 2012.

[50] OWSJ, "What they know." http://blogs.wsj.com/wtk/, 2010.

[51] V. Navalpakkam, L. Jentzsch, R. Sayres, S. Ravi, A. Ahmed, and A. Smola, "Measurement and modeling of eye-mouse behavior in the presence of nonlinear page layouts," in *WWW*, 2013.

[52] Google, "Google ads settings." https://www.google.com/settings/u/0/ads.

[53] "Yahoo! Ad Interest Manager." http://info.yahoo.com/privacy/us/yahoo/opt_out/targeting/, 2012.

[54] J. Rosen, "Who do online advertisers think you are?." http://www.nytimes.com/2012/12/02/magazine/who-do-online-advertisers-think-you-are.html, 2012.

[55] BlueKai, "Bluekai audience data marketplace: Data types." http://bluekai.com/audience-data-marketplace.php, 2014.

[56] BlueKai, "Health and wellness preference data segments." http://www.bluekai.com/health-related-categories.pdf, 2014.

[57] S. Kroft, "The data brokers: Selling your personal information." http://www.cbsnews.com/news/the-data-brokers-selling-your-personal-information/, 2014.

[58] C. Duhigg, "The new york times: How companies learn your secrets." http://www.nytimes.com/2012/02/19/magazine/shopping-habits.html, 2012.

[59] A. Narayanan, "There is no such thing as anonymous online tracking." https://cyberlaw.stanford.edu/blog/2011/07/there-no-such-thing-anonymous-online-tracking, 2011.

[60] B. Krishnamurthy and C. E. Wills, "On the leakage of personally identifiable information via online social networks," in *WOSN*, 2009.

[61] B. Krishnamurthy, K. Naryshkin, and C. Wills, "Privacy leakage vs. protection measures: the growing disconnect," in *W2SP*, 2011.

[62] J. Valentino-DeVries, "Which websites are sharing your personal details?." http://online.wsj.com/news/articles/SB10001424127887324640104578165651354042798, 2012.

[63] AdExchanger, "Ebay pulls the plug on intent data reseller strategy." http://www.adexchanger.com/data-exchanges/ebay/, 2011.

[64] L. Olejnik, C. Castelluccia, and A. Janc, "Why johnny can't browse in peace: On the uniqueness of web browsing history patterns," in *5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2012)*, 2012.

[65] D. Perito, C. Castelluccia, M. A. Kaafar, and P. Manils, "How unique and traceable are usernames," in *PETS*, 2011.

[66] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *IEEE S&P*, 2008.

[67] B. Schneier, "The eternal value of privacy." `http://archive.wired.com/politics/security/commentary/securitymatters/2006/05/70886`, 2006.

[68] G. R. Stone, "Freedom and responsibility." `http://www.huffingtonpost.com/geoffrey-r-stone/freedom-and-responsibilit_b_21308.html`, 2006.

[69] D. J. Solove, "'i've got nothing to hide' and other misunderstandings of privacy," *San Diego Law Review*, vol. 44, p. 745, 2007.

[70] S. Warren and L. Brandeis, "The right to privacy," *Harvard Law Review*, vol. 4, no. 5, 1890.

[71] A. Viseu, A. Clement, and J. Aspinall, "Situating privacy online: Complex perceptions and everyday practices," *Information, Communication & Society*, vol. 7, no. 1, pp. 92–114, 2004.

[72] A. Odlyzko, "Privacy, economics, and price discrimination on the internet," in *ICEC '03*, 2003.

[73] D. Fastenberg, "Facebook firings: Top 10 cases and the nlrb's new guidelines." `http://jobs.aol.com/articles/2011/09/02/facebook-firings-top-ten-cases-and-the-nlrbs-new-guidelines/`, 2011.

[74] T. Gould, "Manager's poor judgment leads to reversal of termination of employee — for poor judgment." `http://www.hrmorning.com/managers-poor-judgment-leads-to-reversal-of-termination-of-employee-for-poor-j` 2013.

[75] D. Kirkpatrick, *The Facebook Effect: The Inside Story of the Company That Is Connecting the World*. Simon & Schuster, 2011.

[76] E. Pariser, "The filter bubble." `http://www.thefilterbubble.com/`, 2012.

[77] D. Gross, "What the internet is hiding from you." http://edition.cnn.com/2011/TECH/web/05/19/online.privacy.pariser/, 2011.

[78] C. Castelluccia and A. Narayanan, "Privacy considerations of online behaviour tracking," *ENISA report*, 2012.

[79] B. Krishnamurthy and C. E. Wills, "Privacy diffusion on the web: a longitudinal perspective," in *Proceedings of the 18th international conference on World wide web*, pp. 541–550, ACM, 2009.

[80] A. Chaabane, M. A. Kaafar, and R. Boreli, "Big friend is watching you: Analyzing online social networks tracking capabilities," in *WOSN*, 2012.

[81] J. Valentino-Devries and J. Singer-Vine, "They know what you're shopping for." http://online.wsj.com/news/articles/SB10001424127887324784404578143144132736214, 2012.

[82] C. Jensen, C. Sarkar, C. Jensen, and C. Potts, "Tracking website data-collection and privacy practices with the iwatch web crawler," in *SOUPS*, 2007.

[83] D. Jang, R. Jhala, S. Lerner, , and H. Shacham, "An empirical study of privacy-violating information flows in javascript web applications," in *CCS*, 2010.

[84] C. Jackson, A. Bortz, D. Boneh, and J. C. Mitchell, "Protecting browser state from web privacy attacks," in *WWW*, 2006.

[85] M. Ayenson, D. J. Wambach, A. Soltani, N. Good, and C. J. Hoofnagle, "Flash cookies and privacy ii: Now with html5 and etag respawning." http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1898390, 2011.

[86] T.-F. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi, "Host fingerprinting and tracking on the web: Privacy and security implications," in *19th Annual Network & Distributed System Security Symposium*, 2012.

[87] Abine, "Donottrackme." https://www.abine.com/index.html, 2014.

[88] InformAction, "Noscript." http://noscript.net/, 2014.

[89] IKRG, "Betterprivacy." https://addons.mozilla.org/en-US/firefox/addon/betterprivacy/, 2014.

[90] D. C. Howe, H. Nissenbaum, and V. Toubiana, "Trackmenot." http://cs.nyu.edu/trackmenot/, 2014.

[91] C. Pederick, "User agent switcher." http://chrispederick.com/work/user-agent-switcher/, 2014.

[92] J. Sullivan, "User agent rg." https://addons.mozilla.org/en-US/firefox/addon/user-agent-rg/, 2014.

[93] "Tor browser." https://www.torproject.org/projects/torbrowser.html.en, 2014.

[94] "Firegloves." http://fingerprint.pet-portal.eu/?menu=6, 2014.

[95] A. Reznichenko, S. Guha, and P. Francis, "Auctions in do-not-track compliant internet advertising," in *CCS*, 2011.

[96] M. Hardt and S. Nath, "Privacy-aware personalization for mobile advertising," in *CCS '12*, 2012.

[97] M. Fredrikson and B. Livshits, "Repriv: Re-envisioning in-browser privacy," in *S&P*, 2011.

[98] I.-H. Hann, K. L. Hui, S.-Y. T. Lee, and I. P. L. Png, "Online information privacy: Measuring the cost-benefit trade-off," in *ICIS*, 2002.

[99] G. Danezis, S. Lewis, and R. Anderson, "How much is location privacy worth?," in *WEIS*, Citeseer, 2005.

[100] D. Cvrcek, M. Kumpost, V. Matyas, and G. Danezis, "A study on the value of location privacy," in *WPES 06*, 2006.

[101] J. P. Carrascal, C. Riederer, V. Erramilli, M. Cherubini, and R. de Oliveira, "Your browsing behavior for a big mac: Economics of personal information online," *arXiv preprint arXiv:1112.6098*, 2011.

[102] A. R. Beresford, D. Kübler, and S. Preibusch, "Unwillingness to pay for privacy: A field experiment," *Economics Letters*, vol. 117, no. 1, pp. 25–27, 2012.

[103] H. Krasnova, T. Hildebrand, and O. Guenther, "Investigating the value of privacy in online social networks: conjoint analysis," in *ICIS*, 2009.

[104] C. Bauer, J. Korunovska, and S. Spiekermann, "On the value of information - what facebook users are willing to pay," in *ECIS*, 2009.

[105] A. Acquisti, L. John, and G. Loewenstein, "What is privacy worth," in *Workshop on Information Systems and Economics (WISE)*, 2009.

[106] J. Gideon, L. Cranor, S. Egelman, and A. Acquisti, "Power strips, prophylactics, and privacy, oh my!," in *SOUPS*, 2006.

[107] J. Grossklags and A. Acquisti, "When 25 cents is too much: An experiment on willingness-to-sell and willingness-to-protect personal information," in *Workshop on Economics of Information Security*, 2007.

[108] B. A. Huberman, E. Adar, and L. R. Fine, "Valuating privacy," *Security & Privacy, IEEE*, vol. 3, no. 5, pp. 22–25, 2005.

[109] A. Acquisti and J. Grossklags, "Privacy and rationality in individual decision making," *Security & Privacy, IEEE*, vol. 3, no. 1, pp. 26–33, 2005.

[110] A. Braunstein, L. Granka, and J. Staddon, "Indirect content privacy surveys: Measuring privacy without asking about it," in *SOUPS*, 2011.

[111] E. Steel, C. Locke, E. Cadman, and B. Freese, "How much is your personal data worth?." http://www.ft.com/intl/cms/s/2/927ca86e-d29b-11e2-88ed-00144feab7de.html.

[112] K. C. Laudon, "Markets and privacy," *Communications of the ACM*, vol. 39, no. 9, p. 92, 1996.

[113] A. Ghosh and A. Roth, "Selling privacy at auction," in *ACM EC*, 2011.

[114] C. Riederer, V. Erramilli, A. Chaintreau, B. Krishnamurthy, and P. Rodriguez, "For sale : your data: by : you," in *HotNets-X*, 2011.

[115] "Google Display Network." http://www.google.com/ads/displaynetwork/, 2011.

[116] C. Doctorow, "Scroogled." http://blogoscoped.com/archive/2007-09-17-n72.html, 2007.

[117] "Google AdSense Help." https://www.google.com/adsense/support/bin/answer.py?hl=en&answer=10528, 2011.

[118] J. Valentino-Devries, "What they know about you." The Wall Street Journal, July 31, 2010.

[119] "Google Ads Preferences." http://www.google.com/ads/preferences/, 2012.

[120] "Google Privacy Policy." http://www.google.com/intl/en/policies/privacy/, 2011.

[121] "Personalized Advertising from Microsoft." http://choice.live.com/AdvertisementChoice/Default.aspx, 2012.

[122] S. Guha, B. Cheng, and P. Francis, "Challenges in measuring online advertising systems," in *Internet Measurement*, 2010.

[123] "Google Adwords Placement Tool." http://adwords.google.com/support/aw/bin/answer.py?hl=en&answer=179238/, 2011.

[124] "Network Advertising Initiative." http://www.networkadvertising.org/, 2011.

[125] "Do Not Track." http://donottrack.us/, 2014.

[126] Google, "The arrival of real-time bidding." http://static.googleusercontent.com/external_content/untrusted_dlcp/www.google.com/pl//doubleclick/pdfs/Google-White-Paper-The-Arrival-of-Real-Time-Bidding-July-2011.pdf.

[127] IDC, "Real-time bidding in the united states and western europe, 2010–2015." http://info.pubmatic.com/rs/pubmatic/images/IDC_Real-Time%20Bidding_US_Western%20Europe_Oct2011.pdf.

[128] A. Acquisti and J. Grossklags, "Privacy and rationality in individual decision making," *Security & Privacy, IEEE*, vol. 3, no. 1, pp. 26–33, 2005.

[129] Google, "Google's cookie matching protocol." https://developers.google.com/ad-exchange/rtb/cookie-guide.

[130] DoubleClick, "Doubleclick ad exchange real-time bidding protocol." https://developers.google.com/ad-exchange/rtb/.

[131] PulsePoint, "Pulsepoint real-time bidding api." http://docs.pulsepoint.com/display/RTB/Real-Time+Bidding+API.

[132] IAB, "Openrtb api specification version 2.1." http://openrtb.googlecode.com/files/OpenRTB-API-Specification-Version-2-1-FINAL.pdf.

[133] PulsePoint, "Real-time bidding protocol request examples." http://docs.pulsepoint.com/display/RTB/Real-Time+Bidding+API#Real-TimeBiddingAPI-BidRequestParameters.

[134] DoubleClick, "Processing the request – example bid request." https://developers.google.com/ad-exchange/rtb/request-guide#example-bid-request.

[135] DoubleClick, "Real-time bidding protocol request examples." https://developers.google.com/ad-exchange/rtb/downloads/realtime-bidding-proto.txt.

[136] S. Yuan, J. Wang, and X. Zhao, "Real-time bidding for online advertising: Measurement and analysis," *arXiv preprint arXiv:1306.6542*, 2013.

[137] Google, "Cpm ads." https://support.google.com/adsense/answer/32725?hl=en.

[138] Google, "Cost-per-click (cpc)." https://support.google.com/adsense/answer/18196?hl=en.

[139] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of finance*, vol. 16, no. 1, pp. 8–37, 1961.

[140] Google, "Decrypting price confirmations." https://developers.google.com/ad-exchange/rtb/response-guide/decrypt-price.

[141] "Httpfox." https://addons.mozilla.org/en-US/firefox/addon/httpfox/.

[142] P. Eckersley, "How unique is your web browser?," in *Privacy Enhancing Technologies*, pp. 1–18, 2010.

[143] PulsePoint, "Private exchange." http://docs.pulsepoint.com/display/RTB/Private+Exchange.

[144] L. Olejnik, M.-D. Tran, and C. Castelluccia, "Selling off privacy at auction," in *NDSS*, 2014.

[145] Yahoo, "Ad interest manager." http://info.yahoo.com/privacy/us/yahoo/opt_out/targeting/.

[146] G. Johnson, "The impact of privacy policy on the auction market for online display advertising." http://gradstudents.wcas.northwestern.edu/~gaj741/GarrettJohnson-JMP.pdf.

[147] Selenium, "Selenium - web browser automation." http://docs.seleniumhq.org/.

[148] Facebook, "Updates to custom audiences targeting tool." http://newsroom.fb.com/News/576/Updates-to-Custom-Audiences-Targeting-Tool.

[149] C. Delo, "Facebook to partner with acxiom, epsilon to match store purchases with user profiles." http://adage.com/article/digital/facebook-partner-acxiom-epsilon-match-store-purchases-user-profiles/239967/.

[150] C. Castelluccia, M. A. Kaafar, and M.-D. Tran, "Betrayed by your ads!," in *PETS*, 2012.

[151] "Collusion." https://www.mozilla.org/en-US/collusion/.

[152] "Targeting & retargeting interview with criteo." http://behavioraltargeting.biz/targeting-retargeting-interview-with-criteo/, 2010.

[153] M. Helft and T. Vega, "Retargeting ads follow surfers to other sites." http://www.nytimes.com/2010/08/30/technology/30adstalk.html, 2011.

[154] Google, "Google: Cookie matching." https://developers.google.com/ad-exchange/rtb/cookie-guide, 2014.

[155] A. Narayanan, "Price discrimination is all around you." http://33bits.org/2011/06/02/price-discrimination-is-all-around-you/, 2011.

[156] AdRoll, "Adroll's ad exchange partners." http://www.adroll.com/about/partners, 2014.

[157] "Criteo gains great results and scale by retargeting audiences through real-time bidding with doubleclick ad exchange." http://doubleclickadvertisers.blogspot.fr/2011/06/criteo-gets-great-results-retargeting.html, 2011.

[158] Criteo, "Criteo to provide customers with access to facebook exchange." http://www.criteo.com/en/news-and-events/press-releases/criteo-provide-customers-access-facebook-exchange, 2012.

[159] Google, "Introducing ad exchange direct deals." http://doubleclickpublishers.blogspot.fr/2011/09/introducing-ad-exchange-direct-deals.html, 2011.

[160] Google, "Preferred deals: A new way to sell in the ad exchange." http://doubleclickpublishers.blogspot.fr/2012/06/preferred-deals-new-way-to-sell-in-ad.html, 2012.

[161] R. Pil, "Preferred deals | fixed priced deals made easy." http://blog.adform.com/real-time-bidding/preferred-deals-fixed-priced-deals-made-easy/, 2013.

[162] Google, "Extra! extra! washington post digital goes programmatic, gets premium rates with doubleclick's ad exchange." http://www.google.com/think/case-studies/wpd-adx.html, 2013.

[163] T. E. Gamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 1, pp. 469–472, 1985.

[164] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *Second Annual International Conference on Mobile and Ubiquitous systems: networks and services*, 2005.

[165] P. Williams and R. Sion, "Single round access privacy on outsourced storage," in *CCS*, 2012.

[166] IBM, "Ibm 4765." http://www-03.ibm.com/security/cryptocards/pciecc/support.shtml, 2014.

[167] S. W. Smith, "Outbound authentication for programmable secure coprocessors," in *ESORICS*, 2002.

[168] "The tor project." https://www.torproject.org/, 2014.

[169] C. Castelluccia, M. A. Kaafar, and M.-D. Tran, "Betrayed by your ads!," in *PETS*, 2012.

[170] Joyent, "Nodejs." http://nodejs.org/, 2014.

[171] Google, "Google ads settings." https://www.google.com/settings/u/0/ads, 2014.

[172] K. Jang, S. Han, S. Han, S. Moon, and K. Park, "Sslshader: Cheap ssl acceleration with commodity processors," in *NSDI*, 2011.

[173] Amazon, "Amazon elastic compute cloud (amazon ec2)." http://aws.amazon.com/ec2/, 2014.

Appendix A: Additive Homomorphic Encryption Scheme from [164]

Description

The main idea of the scheme is to replace the xor (Exclusive-OR) operation typically found in stream ciphers with modular addition (+).

Assume that $0 \leq m < M$. Due to the commutative property of addition, the above scheme is additively homomorphic. In fact, if $c_1 = Enc(m_1, k_1, M)$ and $c_2 = Enc(m_2, k_2, M)$ then $c_1 + c_2 = Enc(m_1 + m_2, k_1 + k_2, M)$.

Note that if n different ciphers c_i are added, then M must be larger than $\sum_{i=1}^{n} m_i$, otherwise correctness is not provided. In fact if $\sum_{i=1}^{n} m_i$ is larger than M, decryption will result in a value m' that is smaller than M. In practice, if $p = max(m_i)$ then M should be selected as $M = 2^{\log_2(p*n)}$.

The keystream k can be generated by using a stream cipher, such as RC4, generated from a private key.

Security Analysis

This additive homomorphic encryption scheme is very similar to a xor-based stream cipher and its security can be proven using a similar proof.

The security relies on two important features: (1) the keystream changes from one message to another and (2) all the operations are performed modulo an integer M. These two features protect the scheme from frequency analysis attacks. In fact, it can be proven that the scheme is *perfectly secure*.

Theorem 1 *The previous encryption scheme is perfectly secure.*

Preuve For plaintext space M, keystream space K, let $\mathcal{K} = |M|$, $m \in [0; M-1]$, $c \in$

Additively Homomorphic Encryption Scheme

Encryption:

1. Represent message m as integer $m \in [0, M-1]$ where M is a large integer
2. Let k be a randomly keystream, where $k \in [0, M-1]$
3. Compute $c = Enc(m, k, M) = m + k \ (mod\ M)$.

Decryption:

1. $Dec(c, k, M) = c - k \ (mod\ M)$

Addition of Ciphertexts:

1. Let $c_1 = Enc(m_1, k_1, M)$ and $c_2 = Enc(m_2, k_2, M)$
2. For $k = k_1 + k_2, Dec(c_1 + c_2, k, M) = m_1 + m_2$

$[0; M-1]$. Set $k^* = c - m \ (mod\ M)$. Then:

$$\begin{aligned} \underset{k \leftarrow \mathcal{K}}{Prob}[Enc(k, m, M) = c] &= \underset{k \leftarrow \mathcal{K}}{Prob}[k + m = c \ (mod\ M)] \\ &= \underset{k \leftarrow \mathcal{K}}{Prob}[k = c - m \ (mod\ M)] \\ &= \underset{k \leftarrow \mathcal{K}}{Prob}[k = k^*] \end{aligned} \tag{1}$$

If we assume that the maximum number of ciphertexts to be added is n and that each plaintext is l-bit long, we must have $M = 2^{l+\log(n)}$, i.e., $|M| = l + \log(n)$. If $c_i = (m_i + k_i)$, then the probability that $c_i \in [0, 2^l - 1]$ is twice the probability that $c_i \in [2^l; M-1]$. More specifically, we have: $\underset{k \leftarrow \mathcal{K}}{Prob}[k = k^*] = 1/(2^l + M)$ if $c > 2^l$ and $\underset{k \leftarrow \mathcal{K}}{Prob}[k = k^*] = 2/(2^l + M)$ if $c < 2^l$.

Since these two equations hold for every $m \in \mathcal{M}$, it follows that for every $m_1, m_2 \in \mathcal{M}$ we have $\underset{k \leftarrow \mathcal{K}}{Prob}[Enc(k, m_1, M) = c] = \underset{k \leftarrow \mathcal{K}}{Prob}[Enc(k, m_2, M) = c]$ which establishes perfect security of the scheme.